Proceedings of the Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT14)

Editors

Markus Dickinson
Erhard Hinrichs
Agnieszka Patejuk
Adam Przepiórkowski

11–12 December 2015
Warsaw, Poland

Sponsors

CLARIN-PL
Common Language Resources and Technology Infrastructure

PAN

Zespół Inżynierii Lingwistycznej

ISBN: 978-83-63159-18-4

Institute of Computer Science
Polish Academy of Sciences
ul. Jana Kazimierza 5
01-248 Warszawa, Poland

http://ipipan.eu/

Programme Chairs

Markus Dickinson, Indiana University, USA
Erhard Hinrichs, University of Tübingen, Germany
Agnieszka Patejuk, ICS Polish Academy of Sciences, Poland
Adam Przepiórkowski, ICS Polish Academy of Sciences, Poland
and University of Warsaw, Poland

Programme Committee

David Bamman, Carnegie Mellon University, USA
Emily Bender, University of Washington, USA
Eckhard Bick, University of Southern Denmark, Denmark
Francis Bond, Nanyang Technological University, Singapore
Claire Bonial, University of Colorado at Boulder, USA
Johan Bos, University of Groningen, The Netherlands
Gosse Bouma, University of Groningen, The Netherlands
António Branco, University of Lisbon, Portugal
Miriam Butt, University of Konstanz, Germany
Stefanie Dipper, Ruhr University Bochum, Germany
Tomaž Erjavec, Jožef Stefan Institute, Slovenia
Eva Hajičová, Charles University in Prague, Czech Republic
Julia Hockenmaier, University of Illinois at Urbana-Champaign, USA
Daniël de Kok, University of Tübingen, Germany
Sandra Kübler, Indiana University, USA
Marie-Catherine de Marneffe, Ohio State University, USA
Detmar Meurers, University of Tübingen, Germany
Jiří Mírovský, Charles University in Prague, Czech Republic
Kaili Müürisep, University of Tartu, Estonia
Joakim Nivre, Uppsala University, Sweden
Stefan Oepen, University of Oslo, Norway
Petya Osenova, Bulgarian Academy of Sciences, Bulgaria
Marco Passarotti, Catholic University of the Sacred Heart, Italy
Marwa Ragheb, Suez Canal University, Egypt
Ines Rehbein, University of Potsdam, Germany
Alexandr Rosen, Charles University in Prague, Czech Republic
Victoria Rosén, University of Bergen, Norway
Djamé Seddah, Paris-Sorbonne University, France
Martin Volk, University of Zurich, Switzerland
Heike Zinsmeister, University of Hamburg, Germany

Local Organising Committee

Adam Przepiórkowski (chair)
Michał Ciesiołka
Konrad Gołuchowski
Mateusz Kopeć
Katarzyna Krasnowska
Agnieszka Patejuk
Małgorzata Włodarczyk
Marcin Woliński
Alina Wróblewska

Preface

The Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT14) is held at the Institute of Computer Science of the Polish Academy of Sciences in Warsaw on 11–12 December 2015. This year's TLT saw 34 paper submissions of which 17 were accepted as long papers and 9 as short papers. Additionally, we are honoured to have distinguished invited speakers: Bonnie Webber (University of Edinburgh, Scotland), with a talk on *Concurrent Discourse Relations: Annotation, Computation and Theory*, and Dag Haug (University of Oslo, Norway), speaking about *Syntactic Discontinuities in Treebanks and Linguistic Theories*.

We are grateful to the programme committee, who worked hard to review the submissions and provided authors with valuable feedback. We would also like to thank CLARIN-PL for sponsoring TLT14, the Institute of Computer Science of the Polish Academy of Sciences for hosting the workshop, and Katarzyna Krasnowska-Kieraś and Marcin Woliński for their help with these proceedings. Last but not least, we would like to wish all participants a fruitful workshop.

Markus Dickinson, Erhard Hinrichs, Agnieszka Patejuk
and Adam Przepiórkowski

Contents

Part I: Long Papers

Part II: Short Papers

Part I

Long Papers

Multiwords, Word Senses and Multiword Senses in the Eukalyptus Treebank of Written Swedish

Yvonne Adesam, Gerlof Bouma and Richard Johansson

Språkbanken, Department of Swedish
University of Gothenburg
E-mail:{yvonne.adesam|gerlof.bouma|richard.johansson}@gu.se

Abstract

Multiwords reside at the intersection of the lexicon and syntax and in an annotation project, they will affect both levels. In the Eukalyptus treebank of written Swedish, we treat multiwords formally as syntactic objects, which are assigned a lexical type and sense. With the help of a simple dichotomy, analyzed vs unanalyzed multiwords, and the expressiveness of the syntactic annotation formalism employed, we are able to flexibly handle most multiword types and usages.

1 Introduction

The *Eukalyptus treebank of written Swedish* will contain about 100.000 tokens and is under active development. It's foremost purpose is to serve as an evaluation corpus for multiple annotation tools, from part-of-speech taggers over sense disambiguators, to parsers. Because of this, it has from the onset been designed with a range of annotations in mind, which has influenced the design of the individual annotation levels. Previous papers [1, 2] have described the purpose of the project and the syntactic annotation of the treebank. In this paper, we focus on the levels of word senses and syntactic structure, which are connected by the shared concern of multiwords. We show how the issue of multiwords and multiword senses is handled by introducing a simple dichotomy in their syntactic annotation. Because both our syntactic annotators and our word sense annotators are confronted with multiwords, we are also able to give an empirical comparison of their annotations.

2 Annotation Levels

The range of annotations in the Eukalyptus treebank can be summarized as follows. Our token definition is roughly the graphic word. Below the token level, we then annotate compound structure; at the token level, lemmata, word senses, parts of speech and morphological features; and above, syntactic structure. When dealing with multiwords, above-token-level annotation also includes multiword lemmata, multiword parts of speech and multiword senses.

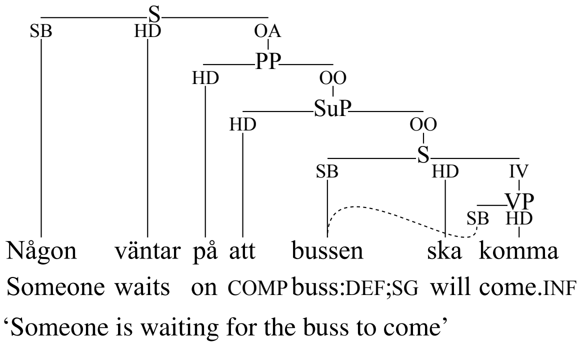
For our inventory of word senses and lemmata, we rely on the SALDO lexical resource [4], which defines senses by placing them in a network of associations. Crucially, SALDO not only contains word senses for single word entries, but, at the time of writing, also for around 8.000 multiword entries, which make up approximately 6.5% of the entries. From the perspective of SALDO, multiword entries are just *word entries*, which means that there is no principled difference in their treatment compared to single word entries. Amongst other things, multiwords are assigned part-of-speech tags in accordance with the regular tag definitions. For instance, there is no concept of 'verb-object idiom' in SALDO, as these are just multiword verbs. For example, the multiword *dra timmerstockar* 'snore' (lit.: 'pull timberloggs') is marked as a multiword verb (VBM), and has *snarka* 'snore' as its primary associative link. Similarly, the expression *lagens långa arm* 'the police' (lit.: 'the law's long arm') is marked as a multiword noun (NNM), with primary link *polis* 'police'. Like SALDO, the Eukalyptus treebank uses parts-of-speech for multiwords. However, in contrast to SALDO and as detailed below, we do, as far as possible, annotate internal syntactic structure in multiwords.

Eukalyptus' syntactic annotation scheme is formally based on the familiar German NEGRA/TIGER scheme [5], combining (possibly discontinuous) phrases with labelled edges for the syntactic functions. A syntactic analysis consists of a primary graph, which is a rooted tree yielding all tokens in the annotation unit, and additional, secondary edges that can be used to express sharing. The combined primary and secondary annotations form an unrestricted directed labelled graph.

We follow, and extend upon, the descriptive traditions of the pioneering annotation guidelines MAMBA [6] from the 1970s and the modern reference grammar *Svenska Akademiens Grammatik* [7]. Phrases in Eukalyptus are generally constrained to be headed by lexical material, and a set of projection rules links the 13 parts-of-speech categories to 10 phrase categories. For each of the 13 parts-of-speech, there is a counterpart multiword part-of-speech, recognizable by a suffix 'M'. However, whereas parts-of-speech formally are terminal node labels in the syntactic tree, multiword parts-of-speech are non-terminal node labels, just like the phrase categories. Non-head children may have one of 20 different grammatical functions, partially depending on the phrase type. An example syntactic tree without any multiwords is given in figure 1.

3 The Analyzed-Unanalyzed Dichotomy: Multiwords as Syntactic Structure

Many types of multiwords have realizations that look like regular syntactic constructions. For instance, a verb-object idiom will take the shape of a non-idiomatic verb object combination, although its variation possibilities may be more or less


S
SB
HD
OA
PP
HD
OO
SuP
HD
OO
S
SB
HD
IV
VP
SB
HD
Någon väntar på att bussen ska komma
Someone waits on COMP buss:DEF;SG will come.INF
‘Someone is waiting for the buss to come’

Note: Apart from the more common abbreviations, the tree uses phrase label SuP for Subordinator Phrase (similar to CP), and dependency labels OA for bound adverbials, OO for (direkt) objekts/complements, and IV for non-finite verbal complements. The solid lines show the primary tree, the dashed line shows the secondary edge used to indicate the implicit subject of *komma* ‘come.INF’.

Figure 1: Syntactic tree for *Någon väntar på att bussen ska komma*.

restricted. From a syntactic annotation perspective, it is attractive to annotate such realizations as regular syntactic structures. The structure may throw light upon some of the regularities we see in the realization, and more importantly, for idioms that allow internal modification, we need the syntactic structure to attach the modifiers in the right place. Consider (1), which involves the multiword *dra timmarstockar* ‘snore’.

(1) den andra slutade dra [$_{\text{NP}}$ de allra tyngsta timmerstockarna]
the other stopped pull.INF the very heaviest timber logs.DEF
‘The other one doesn’t snore as heavily as he did before.’

The determiner *de* and adjectival attribute *allra tyngsta* can only attach to *timmerstockar* if the word is actually allowed to head a phrase and is not just considered part of the multiword.

We might therefore consider multiword annotation to be formally independent of syntactic annotation. At some separate level, we would then represent groups of tokens to which we can attach the multiword senses. However, other types of multiwords pose problems for syntax in ways that suggest that multiwords should be represented directly in syntax. For example, the NP in (2) is headed by what looks like a PP. This would not only be unexpected but it would violate Eukalyptus’ well-formedness rules on heads, which say that heads be lexical and have a part-of-speech related to the phrasal category.

(2) [$_{\text{NP}}$ Anderssons [$_{\text{PP}}$ Till min syster]]
Andersson’s to my sister
‘(Dan) Andersson’s (poem) *For my sister*’

However, if we take into account that the ‘offending’ head is the title of a poem, and can therefore be considered a multiword proper name, we can see that the violation of the well-formedness rules is only apparent: multiword proper names are both lexical and nominal. We can easily adjust our well-formedness criteria to correctly allow (2), if we include information about multiwordhood into the annotation graph.

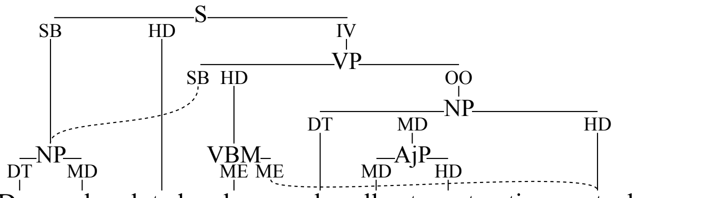
A different problem is found in the multiword proper name in (3), which follows the conventions for person names, but not those for, say, Swedish NPs, without resorting to ad hoc structures. Instead, it appears that it is exactly their grouping as a multiword that allows the multiword elements to participate in the rest of the syntactic structure.

(3) [$_{\text{PN}}$ Jan Johansson] började spela piano 1942.
Jan Johansson started play.INF piano 1942
‘Jan Johansson began to play the piano in 1942.’

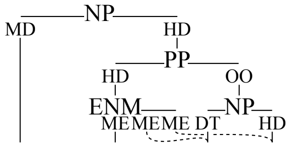
For cases like these, too, we need information about the presence of multiwords and their types as part of the syntactic structure. Without it, we would not be able to assemble the syntactic trees at all.

Eukalyptus therefore integrates multiword annotation into the syntactic annotation, using the possibilities of having secondary edges to be able to ‘overlay’ multiword annotation on top of regular syntactic structures. We recognize two types of multiwords: *Analyzed* multiwords are treated like just indicated: they receive a regular syntactic annotation, and in addition we insert a node with a multiword part-of-speech directly above one of the multiword parts in the primary graph, and link the other multiword parts to these nodes using secondary edges. *Unanalyzed* multiwords, on the other hand, are not considered to have syntactically meaningful internal structure, and their parts are therefore gathered under a multiword node in the primary graph. In both cases, the multiword node serves as the anchor of the SALDO sense id.

The examples in (1) and (2) above contain analyzed multiwords, their trees are given in (4) and (5) below. The special dependency label ME (multiword element) is used for the children of a multiword node. Note that the analyzed multiwords receive a regular syntactic analysis in the primary graph. The additional multiword verb node (VBM) above *dra* in the primary graph (4) can be considered to be superfluous from a syntactic point of view, all it does is provide an anchor for the SALDO id and connect the multiword elements. Since the TIGER/NEGRA formalism does not allow nodes that are only connected with secondary edges, this node has to appear somewhere in the primary graph. But although the multiword proper name node (ENM) above *till* in (5) is without effect in the primary graph directly surrounding it – for instance we still consider the preposition *till* to be the PP’s head – it is instrumental when we check for violations of the headedness rules. In this case, we allow the PP to act as the head of an NP, since it’s yield is also completely under an ENM node (in the full graph).

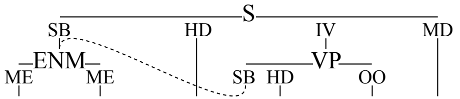


(4) Den andra slutade dra de allra tyngsta timmerstockarna .
the other stopped pull.INF the very heaviest timber logs.DEF
'The other one doesn't snore as heavily as he did before.'



(5) Anderssons Till min syster
Andersson's to my sister
'Andersson's (poem) *For my sister*'

The multiword proper name in (3) is an example of an unanalyzed multiword, its tree is given in (6). Note that in contrast to the previous two examples, the parts of an unanalyzed multiword are children of the multiword node in the primary graph, and the multiword elements are only marked with ME-function.



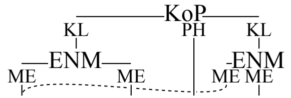
(6) Jan Johansson började spela piano 1942 .
Jan Johansson started play.INF piano 1942
'Jan Johansson began to play the piano in 1942.'

The analyzed-unanalyzed distinction is a type level rather than a token level distinction. As the status of being unanalyzed precludes any modification, and judging modifiability is, in our experience, unreliable, we try to treat as many multiwords as possible as analyzed. Of course, a central property of our scheme is that the choice for syntactical analysis is not mutually exclusive with recognition of its multiword status.

As unanalyzed multiwords we have for example discontinuous coordinators (*både . . . och* 'both . . . and'), circumpositions (*för . . . sedan* 'ago', lit. 'for . . . since'), compound numerals (*sju tusen femhundra* '7500'), phrases of foreign origin (*ad hoc*), and most person names (*Jan Johansson*) and addresses (*Bagaregatan 221B*).

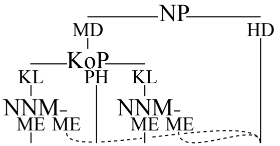
As analyzed multiwords, we may mention adjective-noun combinations (*god man* 'agent with power of attorney', lit.: 'good man'), particle verbs (*gå bort* 'die', lit.: 'go away'), verb-argument idioms (*dra en vals* 'lie', lit.: 'turn a walz'; *gå på gatan* 'prostitute oneself', lit.: 'walk in the street'; *måla fan på väggen* 'assume the worst', lit.: 'paint the devil on the wall'), idiomatic coordinations (*vara ute och cykla* 'be confused/wrong', lit.:'be out and riding a bike'), proverbs (*Äpplet faller inte* *långt ifrån trädet* 'the apple doesn't fall far from the tree'), analyzable proper names of different kinds (*Det sjunde inseglet* 'The seventh seal', *före detta jugoslaviska republiken Makedonien* '(the) former Yugoslav republic (of) Macedonia'), fixed PPs (*före detta* 'former/ex-', lit.:'before this'), NP-formed date expressions (*den fjärde maj* 'the fourth (of) May'), complex prepositions (*på grund av* 'because of', lit.: 'on ground of'), and many more.

Together with the other Eukalyptus annotation principles, our treatment of multiwords is flexible enough to handle a great range of multiword types and uses, including elided multiword parts in coordinations – noted as a problem in [3]. Example (7) shows a coordination of two street addresses, with an elided streetname in the second conjunct. Street addresses are considered unanalyzed multiword proper names (ENM). In coordinations, we may thus see unanalyzed multiwords nodes that dominate some of their elements in the secondary, rather than the primary, graph. Note however, that nowhere in the graph do these elements enter the graph in a non-ME function, which means their inclusion in the graph is only licensed by virtue of their being multiword elements, which is the hallmark of an element in an unanalyzed multiword.



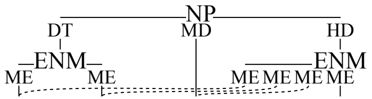
(7) Bagaregatan 221B och 222
Baker street 221B and 222
'Baker street 221B and 222'

The example in (8) shows a coordinated multiword noun (NNM), analyzed as a coordination of adjectival attributes in an NP.



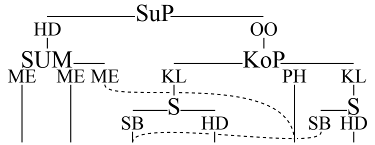
(8) röda och vita blodkroppar
red and white blood cells
'red and white blood cells'

Furthermore, a strength of our approach is that analyzed multiwords can contain other multiwords, thus enabling us to handle embedding of multiwords such as proper names in titles:



(9) Adrian Moles hemliga dagbok
Adrian Mole's secret diary
'The secret diary of Adrian Mole'

Thus far, we have come across one multiword that requires a split analysis, that is, it partially falls into the analyzed class and partially into the unanalyzed class. It concerns the multiword complementizer *vare sig … eller* ‘irrespective of whether … or’ (lit. ‘be.SUBJ REFL … or’). As shown in (10), the first two words together sit in complementizer position (head of subordinator phrase SuP), whilst the last word functions as coordinating conjunction inside the subordinate clause (pseudo-head PH of coordinator phrase KoP).



(10) vare sig du kommer eller går
be REFL you come or go
‘irrespective of whether you are coming or going’

A particular problem that shows up in our treatment of multiwords as syntactic units, and our decision to analyze multiwords and their parts as much as possible, is that multiword elements that do not have an independent usage may require ad hoc analyses. Take, for example, the multiword elements *slint* and *vika* of the idiomatic combinations *slå slint* ‘fail, misfire’ (lit. ‘hit *slint*’) and *ge vika* ‘give way, give in’ (lit. ‘give *vika*’) are not used in other contexts – even though we can easily trace their respective etymologies to the verbs *slinta* ‘to slip’ and *vika* ‘to bend/yield/move’. We have chosen to treat these elements as nouns, because of the existance of other noun-verb pairs in Swedish whose forms relate to each other in the same way, and to treat the complete multiwords as verb-objekt idioms. But since these nouns never occur anywhere else than as (stipulated) objects to these verbs and they do not show object properties like fronting or promotion to subject in a passive, the analysis is not really meaningful. Treating multiwords as tokens, and thus as leaves in the syntactic tree would have avoided this forced classification. However, this would give rise to discontinuous tokens, which may be difficult to handle, visualize and reason about, and, more importantly, it would in essence reduce all multiwords to unanalyzed multiwords. We therefore feel the occasional need for ad hoc analysis is a fair price to pay.

The literature on multiwords, both in theoretical and computational linguistics, consists to a large part in setting up ontologies of multiwords, modelling the syntactic properties of different types of multiwords and investigating consequences for the formal grammar system. Seen against that background, our simple dichotomy may seem to be inadequate as it is nonrestrictive and does not necessarily provide any further insight into the nature of multiwords. However, as part of an annotation scheme, this is not only acceptable but arguably preferable. The task of a syntactic annotation scheme is to allow us to assign the structural distinctions of interest to a broad range of data, rather than to model the language in a generative sense. This is exactly what the analyzed-unanalyzed distinction allows us to do.

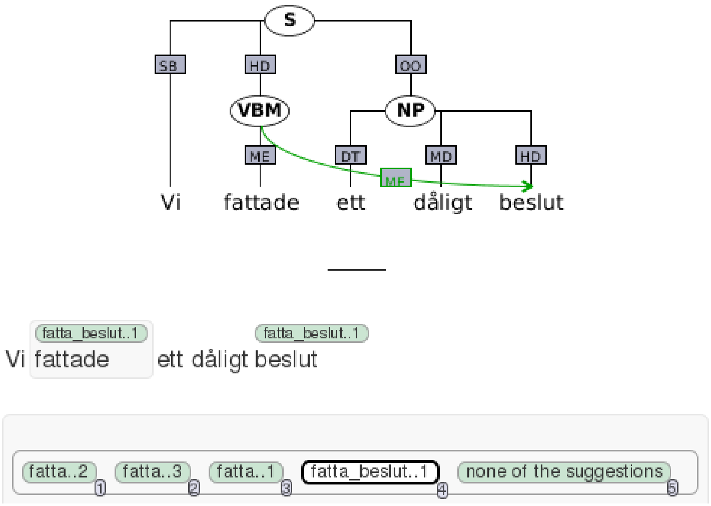


Figure 2: Annotating *vi fattade ett dåligt beslut* ‘we made a bad decision’ in the syntactic task (above) and in the word sense annotation task (below).

4 Multiwords in the Annotation Tasks

As mentioned above, multiwords occur in the word sense annotation as well as in the syntactic annotation. However, sense annotation and syntactic annotation require different annotation tools and methodologies, so for practical reasons we annotate these layers separately. The syntax annotators use a traditional treebank annotation tool,[1] and while their annotation guidelines describe how to treat multiwords, this tool is not integrated with the SALDO lexicon and does not help the annotators decide whether or not a multiword is present in the text. The sense annotators, on the other hand, use a sense annotation tool that is tightly integrated with SALDO, so that for each token, the annotator can choose from a list of single-word and multiword senses defined in SALDO. This makes it easier to know whether the lexicon defines a suitable multiword. We recognize that the subtask of detecting the presence of a multiword is essentially performed in both annotation tasks; however, these annotations will be harmonized in the final stages of the project. It also gives us the opportunity to investigate the influence of our tools and methodologies on this subtask.

Figure 2 shows an example of how a sentence is annotated using the syntactic and word sense annotation tools. In this sentence, *Vi fattade ett dåligt beslut* ‘We made a bad decision’, there is a discontinuous multiword *fatta* . . . *beslut* ‘make . . . decision’, which is annotated on the syntactic level using a node representing the

[1] The syntax tool is based on Synpathy, once developed but no longer maintained at the Max Planck Institute for Psycholinguistics, Nijmegen. See `http://spraakbanken.gu.se/koala` for more information.

multiword verb (VBM). In the word sense annotation tool, the annotator has to pick the multiword sense *fatta beslut*, rather than one of the senses of the single word entry *fatta* ‘grasp; comprehend’.

We compared the multiword annotation in the parts of the treebank where syntactic and sense annotation were both complete; at the time of writing, this part consisted of 7,043 tokens. We did not evaluate how well the annotators were able to make the analyzable/unanalyzable distinction, since this distinction is made on the syntactic level only, nor did we evaluate the actual sense id selected, as this is only part of the sense annotation. The syntactic annotation layer contained 257 multiwords (excluding proper names) in this part of the corpus, while the sense layer had 374 multiwords. In 234 of these cases, the annotation was consistent between the layers, so the syntactic annotations had a precision of 0.91 and a recall of 0.63 with respect to the sense layer. This shows that there are few annotation conflicts: the syntactic annotation is more conservative, which is no doubt caused by the lack of lexicon integration in the syntactic annotation tool, and perhaps also by the required effort of inserting an extra multiword node in the syntactic tree in the case of analyzed multiwords. It is encouraging to see that *when* the syntactic annotators have a strong intuition that a multiword is present, it is also very likely to be annotated as a multiword on the sense level.

We finally considered the multiwords annotated in the sense layer but which were left out in the syntactic layer. As can be expected, they tend to belong to the category of analyzed multiwords, which are often harder to spot and which play a less central role in syntactic annotation. In particular, light verb constructions were often left out by the syntactic annotators (e.g. *fatta beslut* ‘make decision’ or *spela roll* ‘play role’); these are among the syntactically most flexible, and thus inconspicous, of the multiwords.

5 Conclusions

We have shown how the Eukalyptus treebank of written Swedish handles the dual lexical and syntactic nature of multiwords, by formally locating them at the level of syntactic structure. We distinguish between two types of multiwords: analyzed multiwords, whose parts also have a regular syntactic role in the tree; and unalyzed ones, whose parts are only integrated by virtue of being in the multiword.

We are able to compare multiword detection by our lexical and our syntactic annotators. We see that the annotators agree well, however, it is clear that, in terms of tool support, integration of the lexical resource into the syntactic annotation work flow might improve detection of multiwords at that level. Note that, since it is straightforward to mechanically transfer the multiwords found during lexical annotation to the syntactic layer as analyzed multiwords, the lower recall of the syntactic annotators with respect to the lexical annotators is unproblematic. However, an issue for future investigation is how we may improve identification of multiwords that are not currently in the lexicon and are thus likely to be missed in both tasks.

Acknowledgements

The Koala project is funded 2014–2016 by Riksbankens Jubileumsfond, grant number In13-0320:1.

References

[1] Yvonne Adesam, Lars Borin, Gerlof Bouma, Markus Forsberg, and Richard Johansson. Koala – Korp's linguistic annotations. Developing an infrastructure for text-based research with high-quality annotations. In *Proceedings of the Fifth Swedish Language Technology Conference (SLTC)*, Uppsala, november 2014.

[2] Yvonne Adesam, Gerlof Bouma, and Richard Johansson. Defining the Eukalyptus forest – the Koala treebank of Swedish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 1–9, Vilnius, Lithuania, May 2015. Linköping University Electronic Press, Sweden.

[3] Eduard Bejček, Pavel Straňák, and Daniel Zeman. Influence of treebank design on representation of multiword expressions. In Alexander F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 6608 of *Lecture Notes in Computer Science*, pages 1–14. Springer, Berlin, Heidelberg, 2011.

[4] Lars Borin, Markus Forsberg, and Lennart Lönngren. SALDO: a touch of yin to WordNet's yang. *Language Resources and Evaluation*, 47(4):1191–1211, 2013.

[5] Thorsten Brants, Roland Hendriks, Sabine Kramp, Brigitte Krenn, Cordula Preis, Wojciech Skut, and Hans Uszkoreit. Das NEGRA-Annotationsschema. Technical report, Universität des Saarlandes University, Dept of Computerlinguistik, Saarbrücken, 1999.

[6] Ulf Teleman. *Manual för grammatisk beskrivning av talad och skriven svenska*. Studentlitteratur, Lund, 1974.

[7] Ulf Teleman, Staffan Hellberg, and Erik Andersson. *Svenska Akademiens Grammatik*. Svenska Akademien, Stockholm, 1999.

Enriching a Descriptive Grammar with Treebank Queries

Gosse Bouma,[1] Marjo van Koppen,[2] Frank Landsbergen,[3]
Jan Odijk,[2] Ton van der Wouden[4] and Matje van de Camp[5]

[1]University of Groningen, [2]Utrecht University,
[3]Institute for Dutch Lexicography, [4]Meertens Institute, [5]De Taalmonsters
E-mail: g.bouma@rug.nl, j.m.vankoppen@uu.nl,
franklandsbergen@inl.nl, j.odijk@uu.nl,
ton.van.der.wouden@meertens.knaw.nl,
matje@taalmonsters.nl

Abstract

The Syntax of Dutch (SoD) is a descriptive and detailed grammar of Dutch, that provides data for many issues raised in linguistic theory. We present the results of a pilot project that investigated the possibility of enriching the on-line version of the text with links to queries that provide relevant results from syntactically annotated corpora.

1 Introduction

The Language Portal Dutch/Frisian[1] (Landsbergen et al., 2014) is an on-line resource of descriptive linguistic resources, covering syntax, morphology, and phonology of Dutch and Frisian. It contains, among others, an on-line edition of the Syntax of Dutch (SoD) (Broekhuis et al., 2012–), a descriptive grammar of Dutch that goes well beyond the level of detail provided by other sources. Although descriptive, the emphasis in the selection and presentation of phenomena is clearly guided by discussions in the theoretical literature.

In his largely positive review of the SoD volumes on NP syntax, Hoeksema (2013) points out that *"There is a growing body of work in empirical studies of judgment variation [...] that future extensions of this grammar could benefit from, especially when coupled to studies of actual usage patterns in corpus material"* and that *"This particular reader would also have welcomed to see some more lists in the book"*. By enriching the on-line version of SoD with queries over syntactically annotated corpora, the current project tries to accommodate the needs of researchers like Hoeksema.

[1] www.taalportaal.org

Creating a link between a descriptive grammar and a syntactically annotated corpus can be valuable for various reasons. Illustrating a given construction with corpus examples may help to get a better understanding of the variation of the construction and the frequency of these variants. Corpus data may also convince a reader that a given variant actually occurs in (well-formed) text, or in some cases may illustrate that examples judged ungrammatical by the authors of the descriptive grammar do occur with some frequency in actual text.

The (syntactically annotated part of the) Corpus of Spoken Dutch (manually verified, speech from various situations, 1M words) (Oostdijk, 2000), the Lassy Small treebank (manually verified, written material from various genres, 1M words, 65,200 sentences) and the Lassy Large treebank (automatically created[2], written material from various genres, 700M words, 8.6M sentences)) (van Noord et al., 2013) are all suitable corpora for our project. The first two resources provide high-quality data for a limited amount of text, while the last resource provides wide-coverage, but noisy, data. All treebanks follow (with minor modifications) the same annotation standard (Schuurman et al., 2003).

The innovative aspect of this project is the use of syntactically annotated corpora as resource. While descriptive grammars have been based on corpus research, there have been only a few attempts at documenting and extending such grammars with links to relevant examples from treebanks (but see Bender et al. (2012)). The level of annotation that is most valuable for such a resource, i.e. syntactic constituency and grammatical dependency information, does not always align well with the conceptual and ontological assumptions made in the descriptive grammar. Therefore, adding precise treebank queries to a descriptive grammar can be challenging. The goal of the current project is to investigate to what extent a fruitful combination of the two is possible and how much manual effort is required for the development of queries that illustrate phenomena discussed in the descriptive grammar.

Below we describe the treebanks and query tool used in our project. We then give some examples of phenomena that were problematic for our approach, either because annotations did not match, or because the phenomena are so rare that they are hard to find with reasonable precision in the (automatically annotated) treebank. We also give an impression of the coverage of the treebanks, and of the complexity of the queries. Next, we discuss related work and we finish with a discussion of the results.

2 Search interface

We use the web-based corpus query tool PaQu[3] in combination with the example-based query system Gretel[4] for creating and executing treebank queries. The PaQu

[2] using the Alpino parser (van Noord, 2006)

[3] http://zardoz.service.rug.nl:8067/xpath.

[4] http://nederbooms.ccl.kuleuven.be/eng/gretel.

interface returns matching sentences in the selected corpus, with the option to display the matching nodes in the syntactic dependency graph. It displays the query being executed along with a brief description. Queries are dynamic, i.e. the user can switch between treebank corpora, or substitute a given lexical item by an alternative. Furthermore, users can select up to three attributes (i.e. lemma, part of speech, dependency relation, etc.) of matching nodes to obtain a frequency distribution of the attribute-values. Advanced users can also modify the XPATH query as they see fit. Integration of queries into the electronic version of the SoD will be done by adding links (in the form of an icon) to paragraphs and examples for which queries are available.

Construction of queries can be challenging, as it is not always clear how a given constraint should be expressed in terms of XPATH, but also because it is not always clear how a given phenomenon is annotated in the treebanks. To facilitate query formulation, we have used Gretel (Augustinus et al., 2012), a corpus query tool that supports the formulation of XPATH queries that are compatible with the treebank annotation. Users can enter an example sentence, which is parsed automatically by Alpino. Next, relevant parts from the dependency tree can be selected, and a corresponding XPATH query is created. This query can be used to find similar cases in the treebank.

As an example, consider the following statement from SoD concerning the linear order of adjectives and their PP-complements:[5]

> Adjectives typically select a PP as their complement. Although this PP-complement can often either precede or follow the adjective, it is normally assumed that its base-position is the one following the adjective, whereas the pre-adjectival position is derived by leftward movement.
>
> (1) a. Jan was ⟨over deze opmerking⟩ boos ⟨over deze opmerking⟩
> Jan is about that remark angry
> b. Jan is ⟨over zijn beloning⟩ tevreden ⟨over zijn beloning⟩
> Jan is about his reward satisfied

Adjectives selecting for a PP-complement are relatively frequent, and Lassy Small contains many examples of sentences illustrating this syntactic configuration. An example is given in Figure 1. A query that searches the treebank for adjectives selecting a PP-complement is:

[5] http://www.taalportaal.org/taalportaal/topic/link/syntax__Dutch__ap__a2__a2_complementation.2.1.xml.

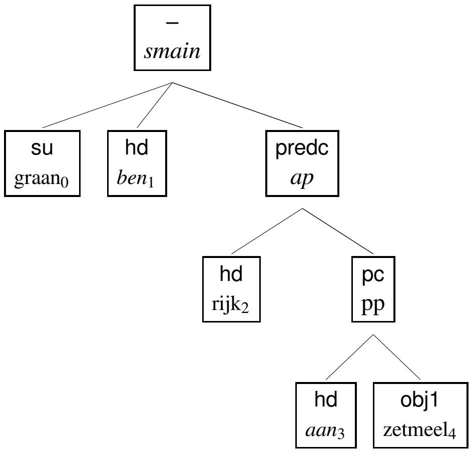


Figure 1: Dependency tree for ’Graan is rijk aan zetmeel’ (*Corn is rich with starch*).

```
//node[@cat="ap"]/
        node[@rel="hd" and
             @pt="adj" and
             ../node[@rel="pc" and
                     @cat="pp"]
            ]
```

This query selects the adjectival head of a node of category AP. Furthermore, the node that matches the head has to have a sibling that is of category PP and whose dependency relation is PC (*prepositional commplement*). Here ’//’ matches an arbitrary position in a tree, ’/’ denotes the ’child of’ relation and ’../’ denotes the sibling relation. The query below adds the constraint that the PP has to precede the adjective:

```
//node[@cat="ap"]/
        node[@rel="hd" and
             @pt="adj" and
             ../node[@rel="pc" and
                     @cat="pp"]/number(@end) = number(@begin)
            ]
```

The attributes `begin` and `end` refer to the begin and end position (in the string) of the corresponding lexical or syntactic node. Here, we require that the end position of the PP has to be equal to the begin position of the adjective.

Counts for adjectives in Lassy Small matching with the first and second query, respectively, are given in Table (2). With 1,125 hits (for 186 lemma’s) PP-complements of adjectives are relatively frequent (i.e. occurring in approx. 2% of the sentences in the corpus). When we restrict attention to PP-A order, however, only 85

Adjective	A+PC	PP-A order
afhankelijk (*dependent*)	100	8
verantwoordelijk (*responsible*)	79	3
afkomstig (*originating*)	56	8
nodig (*needed*)	49	6
eens (*agreed*)	44	18
bezig (*busy*)	34	6
goed (*good*)	34	0
vergelijkbaar (*comparable*)	26	0
bewust (*conscious*)	25	0
tevreden (*content*)	25	0
...		
boos (*angry*)	2	0
total	1,125	85

Table 1: Adjectives with a PP-complement in Lassy Small (second column) and cases where the complement precedes the adjective (third column).

hits remain (for 30 lemma's), i.e. the PP-A order occurs in less than 10% of all cases where we find a PP-complement. This underlines the point made in the descriptive grammar, that A-PP orders are in some sense more basic or less 'marked' than PP-A orders. One might also wonder whether some adjectives do not allow PP-A orders at all. For instance, the adjective *boos*, used in (1-a), does not occur with this word order in Lassy Small. If we execute the same queries on Lassy Large, we find that there are 76 hits for *boos*+PC, but only one for the order PP+*boos*:

(2) Leopold II was over die aantasting van ... bijzonder boos
Leopold II was over that violation of ... extremely angry
Leopold II was extremly upset with that violation of...

This suggests that the PP-A order is exceptional but not impossible for the adjective *boos*.

3 Query development

The SoD uses generic linguistic concepts to present its analyses. Although there is some reference to concepts from generative linguistics, the analyses appear to be general enough to be translatable into most syntactic frameworks. The treebank annotation uses both dependency relations and constituent labels. Dependency relations are widely used in computational linguistics (e.g. see the Universal Dependency format (De Marneffe et al., 2014) that is quickly gaining popularity). The annotation style used in the Dutch treebanks follows earlier work on German (Brants et al., 2003). The dependency annotation allows for crossing branches, something that simplifies annotation of Dutch word order significantly. The preservation of constituent nodes allows a connection with analyses couched in terms of phrase structure trees.

While this set-up suggests that it should be relatively straightforward to translate analyses as formulated in the SoD into treebank terms, in practice this turned out to be challenging for a substantial number of phenomena. This can be due to principled and motivated differences in analysis between the two sources, or by the fact that one of the two sources makes a distinction that is missing in the other.

For instance, the SoD presents a (somewhat artificial) distinction between genitive (3-a) and dative (3-b) nominal complements of adjectives:

(3) a. Jan is zich dat probleem bewust
Jan is REFL that problem$_{ACC}$ aware
John is aware of that problem
b. Het probleem werd Peter niet duidelijk
the problem became Peter$_{DAT}$ not clear
The problem didn't become clear to Peter

In the treebank, the adjective *bewust* does indeed occur with a nominal complement (labeled with the dependency relation *obj1*) (Figure 2, left). Examples like (3-b) occur as well, but not as a single constituent. Instead, *duidelijk* is annotated as predicative complement of the verb *worden* and *Peter* is annotated as an indirect object (*obj2*) complement of *worden* (Figure 2, right).

The most effective method for becoming aware of such mismatches is to parse the example from the descriptive grammar with the example-based query system Gretel (Augustinus et al., 2012). Gretel uses Alpino for syntactic analysis, and thus its results are guaranteed to be consistent with data from the automatically annotated corpus Lassy Large and, given the high level of accuracy and coverage of Alpino, usually also with the manually annotated treebanks. A user can highlight relevant parts of the dependency tree, and Gretel will construct an XPATH query on the basis of this. This query can than be used to search the treebanks for more examples.

While most complementation and modification possibilities mentioned in SoD are easily found in the manually verified treebanks, this is not the case for all word order possibilities being discussed. For instance, the SoD discusses discontinuous APs like (4) in terms of 'PP-over-V', and 'topicalization'.

(4) a. Trots is Jan nooit geweest op zijn vader
Proud has Jan never been of his father
Jan has never been proud of his father
b. Op zijn vader is Jan nooit trots geweest

In the treebank, discontinuous constituents are annotated as such, i.e. as nodes in a dependency graph that dominate a discontinuous part of the sentence (see

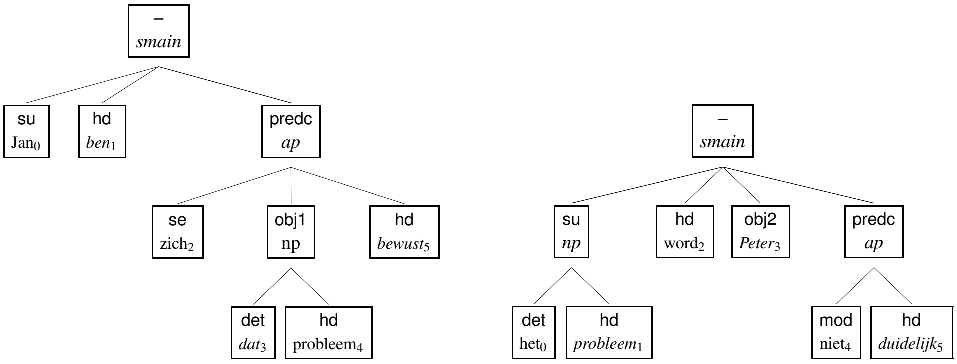


Figure 2: Treebank annotation of *Jan is zich dat probleem bewust* (*John is aware of that problem*) and *Het probleem werd Peter niet duidelijk* (*The problem did not become clear to Peter*).

Figure 3). Using the begin and end attributes of nodes, we can easily search for sentence initial adjectives that have a non-adjacent PP-complement, or, to find cases like (4-b), for sentence initial prepositional complements of adjectives. The second word order occurs with minimal frequency in our data, returning 34 hits on Lassy Small. Some examples are given in (5).

(5) a. Voor deze activiteiten is veel geld nodig
For these activities is much money needed$_{\text{ADJ}}$
These activities require a considerable amount of money
b. Vooral over Mijn vlakke land was Brel zeer tevreden
Especially about Le Plat Pays was Brel very content
Brel was especially pleased with Le Plat Pays
c. Over de oorzaak is nog niets bekend
On the cause is yet nothing known$_{\text{A}}$
Nothing is known yet about the cause

Word orders like (4-a) are far less frequent, however, and can only be found in the Lassy Large treebank. While returning 9 valid hits, search on Lassy Large also returns 11 false or debatable hits. Some examples are shown in (6) below. The last example, (6-d), is a false hit. All false hits are cases of sentences starting with an adjective and ending with a PP, where the parser erroneously prefers to analyse the PP as a complement of a distant adjective instead of attaching it as a modifier to a nearby noun. Despite the moderate accuracy of the automatic annotation on such cases, we believe the result is valuable, as it provides quick access to valid examples that are much harder to find using less sophisticated search methods (i.e. combinations of word and part-of-speech patterns).

(6) a. Verliefd$_{\text{ADJ}}$ was hij doorlopend en dan bij voorkeur [$_{\text{PP}}$ op
love was he continuously and than by preference with

jonge dames tussen 15-20 jaar]
young ladies between 15-20 years
He was continuously in love, and preferably with young ladies in the age of 15-20 years

b. Beroemd$_{\text{ADJ}}$ werd hij [$_{\text{PP}}$ met zijn openlijke uitspraken in de
famous became he with his public statements in the
pers over seks en drugsgebruik]
press on sexuality and drug-use
He became famous for his public statements in the press on sexuality and use of drugs

c. Enthousiast$_{\text{ADJ}}$ werd hij [$_{\text{PP}}$ over de muziek van de jonge
enthousiast became he over the music of the young
componist George Gershwin]
composer George Gershwin
He became enthousiastic about the music of the young composer George Gershwin

d. Beroemd$_{\text{ADJ}}$ is de eerste foto van prinses Beatrix [$_{\text{PP}}$ met Claus
famous is the first picture of princess Beatrix with Claus
von Amsberg]
von Amsberg
The first picture of Beatrix with Claus von Amsberg is famous

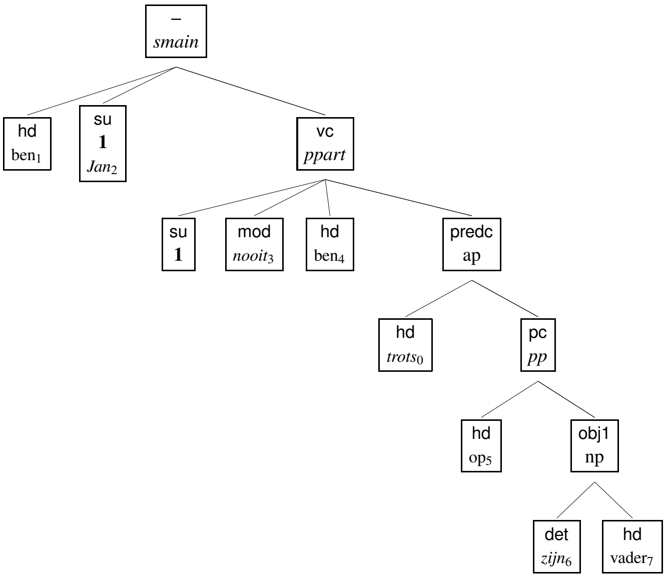


Figure 3: Dependency tree for (4-a). Note that the node dominating *trots op zijn vader* forms a discontinuous constituent.

SoD also discusses PP-A orders in sentence-initial position, like (7).

Corpus	Query type				Sum		%
	Synt (-,+w.o.)		Lex (-,+w.o)				
Lassy Small	228	(168, 60)	527	(409, 118)	755	(577, 178)	63.1
Lassy Large	45	(24, 21)	377	(260, 117)	422	(284, 138)	35.2
CGN	2	(1, 1)	18	(17, 1)	20	(18, 2)	1.7
Total	275	(193, 82)	922	(686, 236)	1,197	(879, 318)	100.0

Table 2: Distribution of queries over corpora

(7) Voor deze functie geschikt is hij niet
For this job suited is he not
He is not fit for the job

Such word orders cannot be found in the manually verified treebanks. In Lassy Large, searching for sentence-initial AP's starting with a prepositional complement also does not return any results. It turns out that the dependency treebank annotation guidelines analyse examples such as (7) as verbal constituents headed by a passive participle.[6] Searching for predicatively used sentence-initial verbal constituents containing a prepositional complement does return a small number of hits.

4 Coverage

For selected sections of the SoD, covering adjectival phrases (complementation, pronominalization, discontinuous cases, modification, and comparative constructions), and adpositions (complementation, absolute PP constructions, and modification), we have constructed almost 1,200 queries.

We assumed that most queries would be formulated over the (manually annotated) Lassy Small corpus, and that the Lassy Large and Spoken Dutch corpus would only be used if Lassy Small returned no hits. Table 2 shows that 63% of the queries indeed use the Lassy Small corpus. The Corpus of Spoken Dutch, eventhough equal in size to the Lassy Small corpus, is only rarely used.

Most queries (922, %) are 'lexical', i.e they search for a specific lexical item occurring in some syntactic context. The other 'syntactic' queries only specify a syntactic context. Queries that do not refer to word order ('-w.o.') are purely configurational. Other queries ('+w.o.') do refer to linear order. By far the most queries are anchored to some lexical item, and also most queries do not refer to linear order. The proportion of lexical queries and the proportion of word-order queries is larger in Lassy Large than in Lassy Small. This suggests that coverage of Lassy Small is sufficient to find examples for many standard syntactic configurations and frequent

[6] It should also be noted that the Alpino parser analyses *geschikt* and similar deverbal adjectives as adjectives. In the conversion step from internal parse representation to treebank annotation, the PoS tag is replaced by a verbal tag.

lexical items, while Lassy Large is used to search for infrequent combinations of a lexical item and syntactic context or word order.

The number of hits per query varies strongly. This is to be expected, as queries that search for some syntactic configuration, without imposing lexical or word order constraints, will usually return a large number of hits. Such queries will be useful mostly because they provide statistics for the syntactic heads occurring in these constructions. Queries that return only a small number of hits, are often queries anchored to a specific lexical item or searching for a non-canonical word order; these are valuable as they illustrate that such constructions do occur, though perhaps rarely, in natural text.

5 Related work

Bender et al. (2012) argue that computational grammars and treebanks can be valuable resources for documenting descriptive grammars. They demonstrate how a descriptive grammar for Wambaya (Nordlinger, 1998) could be used as starting point for the implementation of a computational grammar that covers over 90% of the example sentences in the descriptive grammar and over 75% of held out material from the same language. The computational grammar provides fully explicit analyses of sentences, something that a descriptive grammar cannot do. If the computational grammar is also used (in combination with manual disambiguation decisions to arrive at the optimal parse) to annotate a corpus fragment, a treebank results that can be used to further enrich the descriptive grammar. They argue that 'canned queries' over the treebank may be useful for users who are not familiar with the treebank design or query language, to find exemplars for given syntactic phenomena. If the treebank and query language is adequately documented, users can also formulate their own queries. Our approach provides both options. As Bender et al. (2012) we believe that preformulated queries can be important not only for non-expert users, but also as a means to document the various possibilities of obtaining results from the treebank.

Bender et al. (2012) use the query language Fangorn (Ghodke and Bird, 2012). van Noord et al. (2013) show that XPATH queries over Alpino-style dependency trees (where there is a one-to-one correspondence between linguistic dominance and embedding of elements in XML, and where word order is encoded by XML attributes that register string positions) can deal with all the cases used as test cases for linguistic query languages by Lai and Bird (2004). We therefore prefer to use XPATH, as it has the important additional advantage that it is a widely accepted standard supported by numerous XML processing tools.

Hashimoto et al. (2008) use an annotated treebank to obtain detailed syntactic information on the lexical types that occur in the treebank. Their aim is to ensure consistency both in future extensions of the treebank, as well as for computational grammars that follow the annotation guidelines underlying the treebank annotation. Flickinger et al. (2014) similarly use a treebank primarily as a means for documenting and validating their computational lexicon and grammar. Our work differs in that it uses a treebank to enrich a descriptive grammar that is completely unrelated to the treebank or the guidelines used for annotating the treebank. As a consequence, we cannot assume a transparant conceptual mapping between analyses as discussed in the descriptive grammar on the one hand and underlying the treebank annotation on the other.

6 Conclusions

After completion of approx. 1,200 queries, we have learned that creating suitable queries for a given fragment from the SoD requires creativity and careful experimentation, tuning, and documentation. Construction of queries is far from deterministic, that is, different annotators will have different opinions concerning the most suitable query for a given example or phenomenon. In a substantial number of cases, there are mismatches (in constituent structure, in part-of-speech) between the presentation in the SoD and the treebank annotation. While this makes the development of queries harder, it also underlines the value of the current project: by systematically exploring the way various linguistic examples are annotated in the treebank, we provide a starting point for further corpus exploration for users that have a general linguistic interest but who are not necessarily experts on Dutch treebank annotation.

The manually verified treebanks almost always provide sufficient examples of basic word order patterns for queries that are not restricted to a specific adjective or preposition. For queries that search for a specific lexical head or for less frequent word order patterns, the Lassy Large treebank usually has to be used. In that case, users must be prepared to see also a certain number of false hits. However, there are also examples in the SoD that cannot be found in a 700M word corpus. The conclusion that such word orders are not found in the language would be too strong, but it might be a starting point for further research (i.e. *does this construction occur only in certain registers or discourse settings?*) or for an alternative analysis (i.e. *do these cases really involve adjectives?*).

References

Liesbeth Augustinus, Vincent Vandeghinste, and Frank Van Eynde. Example-based treebank querying. In *LREC*, pages 3161–3167, 2012.

Emily M. Bender, Sumukh Ghodke, Timothy Baldwin, and Rebecca Dridan. From database to treebank: On enhancing hypertext grammars with grammar engineering and treebank search. *Language Documentation & Conservation*, Special Publication No. 4:179–206, 2012.

Thorsten Brants, Wojciech Skut, and Hans Uszkoreit. Syntactic annotation of a German newspaper corpus. In *Treebanks*, pages 73–87. Springer, 2003.

H. Broekhuis, E. Keizer, M. den Dikken, N. Corver, and R. Vos. *Syntax of Dutch*. Amsterdam University Press, Amsterdam, 2012–. (several volumes).

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of LREC*, pages 4585–4592, 2014.

Dan Flickinger, Emily M Bender, and Stephan Oepen. Towards an encyclopedia of compositional semantics: Documenting the interface of the English Resource Grammar. In *Proceedings of the Ninth International Conference of Language Resources and Evaluation (LREC-2014)*, Reykjavik, Iceland, 2014.

Sumukh Ghodke and Steven Bird. Fangorn: A system for querying very large treebanks. In *COLING (Demos)*, pages 175–182, 2012.

Chikara Hashimoto, Francis Bond, Takaaki Tanaka, and Melanie Siegel. Semi-automatic documentation of an implemented linguistic grammar augmented with a treebank. *Language Resources and Evaluation*, 42(2):117–126, 2008.

Jack Hoeksema. Review of: Syntax of Dutch. Noun and Noun Phrases vols. 1 and 2. *Lingua*, 133:385–390, 2013.

Catherine Lai and Steven Bird. Querying and updating treebanks: A critical survey and requirements analysis. In *Proceedings of the Australasian language technology workshop*, pages 139–146, 2004.

Frank Landsbergen, Carole Tiberius, and Roderik Dernison. Taalportaal: an online grammar of Dutch and Frisian. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2206–2210, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA).

Rachel Nordlinger. *A Grammar of Wambaya*. PhD thesis, Research School of Pacific and Asian Studies, The Australian National University, Canberra, 1998.

Nelleke Oostdijk. The Spoken Dutch Corpus: Overview and first evaluation. In *Proceedings of LREC 2000*, pages 887–894, 2000.

Ineke Schuurman, Machteld Schouppe, Heleen Hoekstra, and Ton van der Wouden. CGN, an annotated corpus of spoken Dutch. In *Proceedings of 4th International Workshop on Language Resources and Evaluation*, pages 340–347, 2003.

Gertjan van Noord. At Last Parsing Is Now Operational. In Piet Mertens, Cedrick Fairon, Anne Dister, and Patrick Watrin, editors, *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, pages 20–42. 2006.

Gertjan van Noord, Gosse Bouma, Frank van Eynde, Daniel de Kok, Jelmer van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste. Large scale syntactic annotation of written Dutch: Lassy. In Peter Spyns and Jan Odijk, editors, *Essential Speech and Language Technology for Dutch: the STEVIN Programme*, pages 147–164. Springer, 2013.

Semantic Role Annotation
in the Ancient Greek Dependency Treebank

Giuseppe G. A. Celano and Gregory Crane

Department of Computer Science, Digital Humanities
University of Leipzig
E-mail: {celano|crane}@informatik.uni-leipzig.de

Abstract

The article presents an annotation scheme for semantic role annotation for the Ancient Greek Dependency Treebank. It consists in a hierarchical tagset which implements H. W. Smyth's *A Greek Grammar for Colleges*, partly revised to meet algorithmic adequacy. The results are then shown for the intercoder agreement values calculated for two annotators who have treebanked a pilot corpus containing 417 sentences (6486 tk).

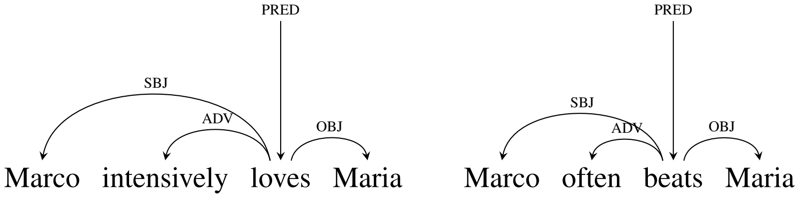
1 Introduction

Semantic role (SR) labeling consists in identifying the participants of an event, by rendering explicit who does what in what circumstances (for a recent linguistic introduction see Haspelmath and Hartmann [15], Levin and Rappaport Hovav [19], and references therein). Look at the following examples:

(1) Marco intensively loves Maria

(2) Marco often beats Maria

Example (1) and (2) show the same morphosyntactic analysis. The PoS of the words of both sentences are NNP, RB, VBZ, and NNP respectively (Santorini [23]). Similarly, the dependency tree structure (Hajičová et al. [13]; Cinková [10]), as well as the syntactic labels, are identical:



Even though the morphosyntactic analysis for both sentences is the same, their meaning is very different. This is captured by their SR analyses:

(3) Marco intensively loves Maria
EXPERIENCER DEGREE V STIMULUS

(4) Marco often beats Maria
AGENT TIME V PATIENT

As the analyses for Example (3) and (4) illustrate, the sentences have very different meanings: in (3), for example, "Marco" is an experiencer, while in (4) he is an agent; similarly, "Maria" is a stimulus in (3), but a patient in (4). Being able to enrich a treebank with SR labels would allow automatic extraction of the shallow semantics of sentences, from which a great number of NLP tasks, such as machine translation and document classification, as well as linguistic research, could enormously benefit.

There currently exist two major influential treebank projects which are particularly relevant for our SR annotation in the Ancient Greek Dependency Treeebank (AGDT): PropBank and the Prague Dependency Treebank.

PropBank (Kingsbury and Palmer [17]; Babko-Malaya [1]) adds manual argument structure annotation to the phrase structure annotation of the Penn Treebank. PropBank is now linked in SemLink (Palmer [22]) to other manually annotated semantic resources, i.e., Verbnet, Framenet, and WordNet. Relying on Verbnet and Framenet, the AMR project (Banarescu et al. [4]) is attempting to build a "Sembank", which provides a detailed sentence semantic description.

The Prague Dependency Treebank (PDT) has developed the most detailed guidelines for the manual annotation of semantic roles for both Czech and English (PCEDT) (Mikulová et al. [21]; Cinková et al. [11]). Since in the PDT and the PCEDT fine-grained SRs not only for arguments but also for adjuncts are annotated, these resources currently provide the richest treebank annotation for both Czech and English SRs.

One serious limitation of both the PDT and the PCEDT, as well as ProbBank, is that the semantics of the first and second verb argument is not investigated, in that semantically empty (or macrorole) labels are employed (ARG0 and ARG1; ACT and PAT) for the subject and object. Verbnet could be helpful to mark such SRs, but this possibility, to the best of our knowledge, has not yet been explored.

In the following sections the SR annotation for the Ancient Greek Dependency Treebank (AGDT) will be presented. Section (2) outlines the SR annotation scheme for the AGDT. Section (3) presents the ongoing phase of creation of a semantically annotated corpus for AG. Section (4) contains concluding remarks.

2 A semantic role annotation scheme for the AGDT

The AGDT is the oldest (ongoing) treebank project for Ancient Greek (Bamman and Crane [3]; Bamman et al. [2]). It originally contained 20 texts (374.490 tk) which have been annotated for morphology and syntax, following syntactic guidelines that much rely on the ones developed for the analytical level of the PDT. Currently it has been expanded to comprise 32 texts (558.123 tk).

Other AG texts have been annotated within the PROIEL project (Haug and Jøhndal [16]), whose annotation scheme is very similar to that of the AGDT. Notably, the PROEIL corpus also contains information structure annotation.

Since 2014 more detailed morphosyntactic guidelines have been made available and a new semantic annotation layer has been introduced for the AGDT (Celano [8]). The latter consists in the identification of SRs according to the description provided by H. W. Smyth's Greek grammar ([24]). As is well known, a great number of issues arise when designing/choosing an annotation scheme, especially for semantics (Bunt et al. [5]; Bunt [6]; Kübler and Zinsmeister [18]; Flickinger et al. [12]), among which are the following:

- annotation goal
- informativity vs. specificity
- scalability
- building on related resources/standards

We aimed to enrich the existing morphosyntactic annotation with SR information that could be easily understood and therefore annotated by classicists. Grammar teaching for AG is currently highly homogeneous across countries, relying on "national" grammars directly or indirectly derived from early twentieth-century monumental works, such as *Ausführliche Grammatik der griechischen Sprache* by R. Kühner and B. Gerth. We therefore decided to adopt as our model the most notorious English offspring of such grammars, i.e., Smyth's *A Greek Grammar for Colleges* (SG) [24].

SG was designed for college students as a tool to study AG and, at the same time, as a quick reference grammar. It represents a well-balanced compromise between an all-comprehensive grammar treatise and a study grammar. Relying on SG description, a hierarchical tagset was designed which, on the basis of the PoS annotation at the morphological layer, allows an annotator to get to a SR annotation in a guided way.

Figure (1) shows a part of the SR annotation algorithm. The *dative of interest*, for example, allows subcategorizations such as the *dative of the possessor*, the *dative of advantage*, and the *dative of disadvantage*. In turn, the *dative of interest* is in a mutually exclusive path with, for example, the *dative of relation*. The full path allowing the annotation of, for example, a *dative of possessor* is: *dative* > *dative proper* > *dative of interest* > *dative of the possessor*. The first step is a PoS with its features (e.g., a dative noun) annotated at the morphological layer, while the followings steps represent a gradient SR annotation (the full algorithm can be inspected at Celano [7], which details the relationship between all the categories involved and their definitions).

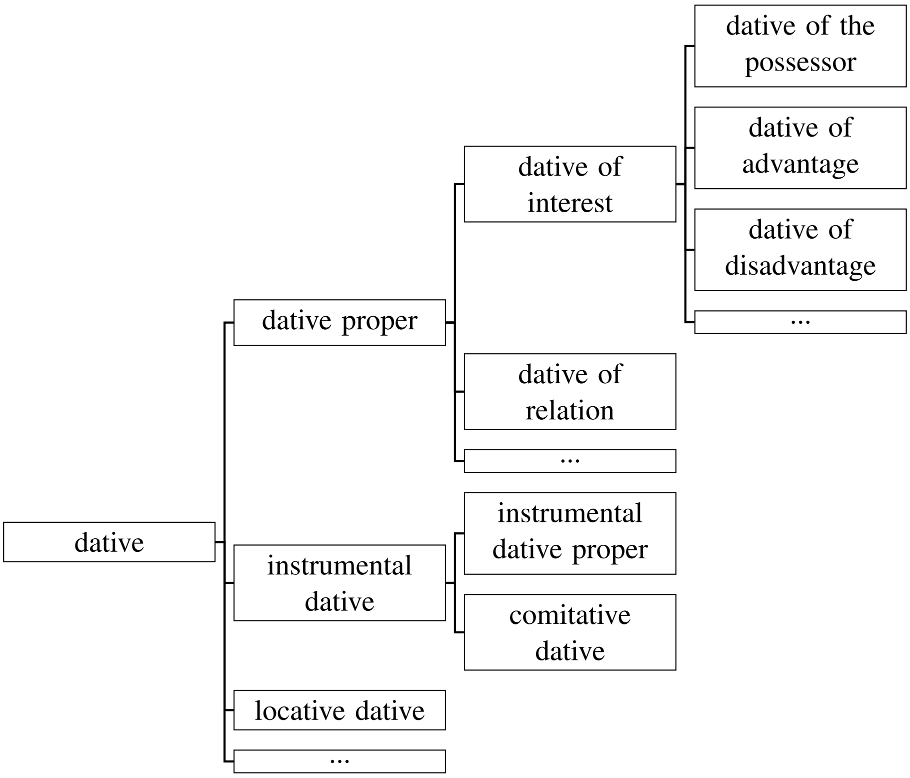


Figure 1: Tree diagram for the annotation of the dative (excerpt)

The annotation algorithm allows labeling of a great number of SRs. They are the ones identified in SG, revised/augmented in the light of more recent literature. For example, the SR for time can be further annotated according to the fine-grained categories identified by Haspelmath [14]: *simultaneous location*, *sequential location*, *sequential-durative location*, *temporal distance*, and *temporal extent*. It is currently possible to annotate 39 SRs. Similarly to PropBank and the PCEDT, it is not currently possible to annotate in the AGDT the SRs expressed by the nominative and the accusative. Figure (2) shows the inventory of SRs which can be annotated.

One of the major advantages in adopting the categories identified in an already existing grammar, such as SG, is that one can avoid the difficult and time-consuming task of defining each concept. In the guidelines developed for the annotation of these SRs, therefore, each of them is only introduced, in that they are all hypertextually linked to the relevant sections in SG, where each annotator can access full definitions and plenty of examples, which turn out to be particularly useful in annotating non-prototypical instances (see Celano [8] for full documentation, including links, definitions, and examples). In general, building on an already existing grammar allowed the definition of an annotation scheme being not only more easily understandable to annotators but also detailed and very well docu-

semantic role	semantic role
accompaniment	(place) separation
accompanying circumstance	(place) terminal location/direction
advantage	possession or belonging/possessor
disadvantage	price and value
agent	purpose
friendly association	quality
hostile association	relation
cause	recipient or addressee
conformity	reference
connection	respect
crime and accountability	feeling
(distinction and) comparison	(time) simultaneous location
explanation	(time) sequential location
instrument or means	(time) sequential-durative
manner	(time) temporal distance
material or contents	(time) temporal extent
measure	topic
measure of difference	source
(place) location	standard of judgment
(place) path	

Figure 2: Semantic roles in the AGDT

mented, which is difficult to achieve from scratch.

The derivation of the algorithm from SG has sometimes required remodeling of the categories available. For example, although the dative of space and time is fully treated under the *locative dative* (SG 351-353), a not well-defined variant of it (SG 350) is subsumed under the *comitative dative* (which is in a mutually exclusive relationship with the *locative dative*). We brought the underspecified comitative variant under the *locative dative*.

3 Annotating semantic roles

Our SR annotation scheme has so far been employed to annotate 50 Aesopian fables and the passage 1.1.1-1.4.1 from Apollodorus' *Bibliotheca* (in total, 417 sentences, 6486 tk). The texts were assigned to two (one undergraduate and one graduate) independent ERASMUS students in Classics, who annotated the morphology, syntax, and semantics of the texts over a three-month time span.

They were intensively trained by an expert instructor for two weeks, during which they acquired the basics of treebanking and annotation (one annotator already had some previous knowledge of the morphosyntactic annotation). After

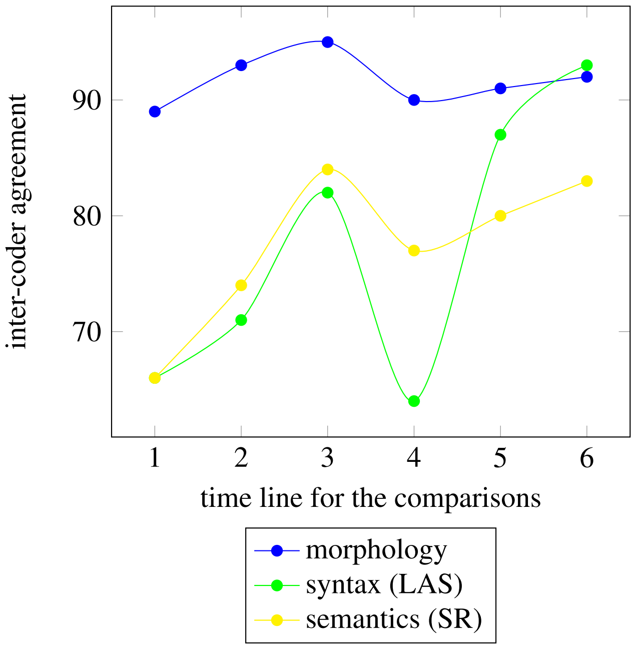


Figure 3: Inter-coder agreement for the last texts in the AGDT

that, they started to annotate, and their annotations were checked regularly by calculating chance-corrected (Cohen's Kappa) inter-coder agreements (ICA).

After each calculation the results were discussed together with the instructor: whenever clear annotation mistakes were identified, the annotators were asked to correct them. The following plot shows the values for all the inter-coder agreements calculated.

The plot displays the performance of the two annotators over time. The results for the six inter-coder agreements were calculated sequentially on different (portions of the) texts. It is interesting to note that the 4th comparison is the only one concerning Apollodorus' text: the ICA for syntax is very low probably because of the high number of long and complex coordination structures, which are peculiar to this text (cf. Kübler [20]).

The average ICA value for the SR annotation from the 3rd comparison is 81%. This is comparable to those achieved for the PCEDT (best result for the tectogrammatical layer for English is 85,7%; Cinková et al. [11]) and for the annotation for topic-focus articulation in the PDT 2.0 (between 80-90%; Veselá et al. [25]). The texts have been incorporated into the AGDT and are available online [9]

4 Conclusion

In the present paper we have presented the SR annotation available for the AGDT. We have outlined the design of the annotation tagset, which is particular in being hierarchical. We have then shown the results for the ICA values calculated over a three-month time span for two student annotators.

Considering that the annotators had no previous knowledge of semantic annotation and the time of their study and training was relatively short, we believe that the (relatively) high ICA values (from the 3rd comparison onwards) for such a complex SR annotation may be due to the structure of the tagset, which, in being hierarchical, constrained the annotators' choices and thus arguably determined more consistency and agreement. This conclusion should, however, be tested with an ad hoc experiment consisting in direct comparison of both hierarchical and non-hierarchical tagsets, which was outside the scope of the present research.

References

[1] Babko-Malaya, Olga (2005) *PropBank Annotation Guidelines*. (URL: http://verbs.colorado.edu/~mpalmer/projects/ace/PBguidelines.pdf).

[2] Bamman, David, Francesco Mambrini, and Gregory Crane (2009) An Ownership Model of Annotation: the Ancient Greek Dependency Treebank. In *the 7th International Workshop on Treebanks and Linguistic Theories*, Groningen, Netherlands.

[3] Bamman, David and Gregory Crane (2001) The Ancient Greek and Latin Dependency Treebanks. In Sporleder, Caroline, den Bosch, Antal and Zervanou Kalliopi (eds.) *Language Technology for Cultural Heritage*. Berlin: Springer.

[4] Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider (2013) Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, Sofia, Bulgaria.

[5] Bunt, Harry, Johan Bos, and Stephen Pulman (2014) Computing Meaning: Annotation, Representation, and Inference. In Bunt, Harry, Johan Bos, and Stephen Pulman (eds.) *Computing Meaning*. Dordrecht: Springer.

[6] Bunt, Harry (2014) Annotations that Effectively Contribute to Semantic Interpretation. In Bunt, Harry, Johan Bos, and Stephen Pulman (eds.) *Computing Meaning*. Dordrecht: Springer.

[7] Celano, Giuseppe G. A. (2014) *A Semantic Role Tagset for the Ancient Greek Dependency Treebank*. (URL: http://services.perseids.org/arethusa-configs/smyth3.json).

[8] Celano, Giuseppe G. A. (2014) *Guidelines for the Ancient Greek Dependency Treebank 2.0*. (URL: https://github.com/PerseusDL/treebank_data/blob/master/AGDT2/guidelines/Greek_guidelines.md).

[9] Celano, Giuseppe G. A., Gregory Crane, and Bridget Almas (2015) *Ancient Greek Dependency Treebank 2.0*. (URL: https://github.com/PerseusDL/treebank_data/tree/master/v2.0/Greek).

[10] Cinková, Silvie, Jan Hajič, Marie Mikulová, Lucie Mladová, Anja Nedolužko, Petr Pajas, Jarmila Panevová, Jiří Semecký, Jana Šindlerová, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský (2006) *Annotation of English on the Tectogrammatical Level*. (URL: https://ufal.mff.cuni.cz/techrep/tr35.pdf).

[11] Cinková, Silvie, Josef Toman, Jan Hajič, Kristýna Čermáková, Václav Klimeš, Lucie Mladová, Jana Šindlerová, Kristýna Tomšů, and Zdeněk Žabokrtský (2009) Tectogrammatical Annotation of the Wall Street Journal, *Prague Bulletin of Mathematical Linguistics*, 2009, 92, pp. 85-104.

[12] Flickinger, Dan, Stephan Oepen, and Emily M. Bender. *Sustainable Development and Refinement of Complex Linguistic Annotations at Scale* (forthcoming)

[13] Hajičová, Eva, Zdeněk Kirschner, and Petr Sgall (1999) *A Manual for Analytic Layer Annotation of the Prague Dependency Treebank* (English translation). ÚFAL MFF UK, Prague, Czech Republic.

[14] Haspelmath, Martin (1997) *From Space to Time: Temporal Adverbials in the World's Languages*. Munchen: LINCOM Europa.

[15] Haspelmath, Martin and Irene Hartmann (2015) *Comparing Verbal Valency Across Languages*. (URL: http://www.academia.edu/9213082/ Comparing_verbal_valency_across_languages_with_Iren_Hartmann_).

[16] Haug, Dag T. T. and Marius L. Jøhndal (2008) Creating a Parallel Treebank of the Old Indo-European Bible Translations. In Sporleder, Caroline and Kiril Ribarov (eds.) *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, pp. 27-34.

[17] Kingsbury, Paul and Martha Palmer (2002) From Treebank to PropBank. In *Third International Conference on Language Resources and Evaluation, LREC-02*, May 28- June 3, 2002, Las Palmas, Canary Islands, Spain.

[18] Kübler, Sandra and Heike Zinsmeister (2015) *Corpus Linguistics and Syntactically Annotated Corpora*. London: Bloomsbury.

[19] Levin, Beth and Malka Rappaport Hovav (2005) *Argument Realization*. Cambridge: CUP.

[20] Maier, Wolfgang, Sandra Kübler, Erhard Hinrichs, and Julia Krivanek (2012) Annotating Coordination in the Penn Treebank. In *Proceedings of the 6th Linguistic Annotation Workshop (LAW)*, Jeju Island, Korea.

[21] Mikulová, Marie et al. *Annotation on the Tectogrammatical Layer in the Prague Dependency Treebank: Annotation Manual*. (URL: http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/t-layer/pdf/t-man-en.pdf).

[22] Palmer, Martha (2009) Semlink: Linking PropBank, VerbNet, and FrameNet. In *Proceedings of the Generative Lexicon Conference*, Pisa, Italy.

[23] Santorini, Beatrice (1990) *Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision, 2nd Printing)*. (URL: https://catalog.ldc.upenn.edu/docs/LDC99T42/tagguid1.pdf).

[24] Smyth, Herbert Weir (1920) *A Greek Grammar for Colleges*. New York: American Book Co.

[25] Veselá, Kateřina, Jiří Havelka, Eva Hajičová (2004) Annotators' Agreement: the Case of Topic-Focus Articulation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation, European Language Resources Association*, Lisboa, Portugal, 2004, pp. 2191-2194.

A Grammar-Book Treebank of Turkish

Çağrı Çöltekin

Department of Linguistics
University of Tübingen
E-mail: ccoltekin@sfs.uni-tuebingen.de

Abstract

This paper introduces a new Turkish dependency treebank following the Universal Dependencies annotation scheme. The treebank is built on example sentences from a grammar book, which cover a wide range of the linguistic constructions. Thus, the resulting treebank is a valuable resource for theoretical (linguistic) research as well as testing computational tools for the coverage of the constructions found in the language.

1 Introduction and motivation

Common choices of source material for treebanks include news corpora from a single source [12, 21], random sentences from the Web and other freely available sources [17, 22], or from sentences balanced across a selected set of document categories [2]. Although these treebanks are useful for the purpose they are created for, and they may be representative of the language use to some degree, it is unlikely that they include infrequent grammatical constructions because of the power laws that govern the distribution of linguistic constructions at many levels.

The aim of the present work is to cover a large set of morpho-syntactic constructions with a minimal amount of annotation effort. To this end, comprehensive grammar books provide an excellent source of sentences, since they are selected by the authors to cover all constructions in the language, including infrequent but interesting ones. Such books are also more likely to cover examples of spoken and non-standard language use in comparison to most treebanks that are based on written, and possibly carefully edited, language material.

Our initial motivation for constructing the present treebank has been to set annotation guidelines for Turkish for the Universal Dependencies (UD) project [1]. However, such a treebank can be useful for many other purposes. For example, it is a valuable resource for checking existence of certain features or syntactic constructions in the language. Therefore, it may be useful in (theoretical) linguistic studies, including cross-linguistic comparisons. The rich linguistic descriptions in the source grammar book (e.g., glosses and detailed descriptions that accompany the example sentences) make the use of the treebank even more practical. Researchers can always refer to the original verbal description of the sentence in the grammar. Furthermore, it could be used for testing and qualitative evaluation of parsers, as one can observe type of errors that are difficult to encounter in typical test sets used for parser evaluation. Although the present treebank would not be appropriate as the only training data for parsers, it may improve parser performance by providing the data for infrequent constructions if the treebank is used as additional training data. For both purposes, a well-documented annotation standard is important.

Currently the most prominent treebank of Turkish is the METU-Sabancı treebank [2, 13], which also sets the de facto standard for dependency annotation of Turkish. The treebank contains a selection of sentences from the METU corpus [15] which is built as a balanced corpus across a number of different domains. The METU-Sabancı treebank is relatively small in comparison to the treebanks available for other languages (5 635 sentences and 56 424 tokens, in comparison to approximately 100 000 or more sentences usual in today's treebanks [17, 21]). The treebank has not been updated since its first release in 2003, and annotation errors and inconsistencies are frequently reported in the literature [6, 19, 9, 16]. Some of these studies also report improvements to the annotation scheme and individual annotations. However, except modifications by Seeker and Çetinoğlu [16] breaking the cycles in the dependency graphs, these improvements have not yet been released. The METU-Sabancı treebank is also converted to UD scheme as part of HamleDT [25], through an automatic process.

Besides METU-Sabancı treebank, other Turkish treebank constructions efforts include automatically or semi-automatically constructed Swedish-Turkish [3] and English-Turkish [24] parallel treebanks, and a small LFG treebank of 32 sentences in the INESS project [14]. The examples of the use of descriptive linguistic information for enriching NLP resources in earlier literature include [23, 4].

The study presented here differs from the earlier work by manually annotating a selection of sentences covering a wide range of constructions in the language. The annotations in the treebank follow the current UD annotation scheme (version 1.2) as closely as possible. In this paper, we focus on introducing the treebank, and discussing some of the issues in the dependency annotation of Turkish. Special attention is paid to divergences from the UD annotation scheme, and differences from the METU-Sabancı treebank.

2 Treebank and the annotation procedure

The treebank consists of 2 803 example sentences or sentence fragments extracted from a recent comprehensive grammar of Turkish by [10]. 410 of the treebank entries are sentence fragments, e.g., example noun phrases. For the rest of this document, we refer to all entries in the treebank, as 'sentences'.

The average length of the sentences in the treebank is shorter than sentences found in typical treebanks. The treebank consist of 16 516 surface tokens (5.89 per sentence, cf. 10.01 in METU-Sabancı treebank). The number of syntactic tokens, or inflectional groups (see Section 3.1 for details of tokenization), is 18 146, with a ratio of 1.10 syntactic tokens per surface token. This number is lower than the METU-Sabancı treebank (1.20) because of the more conservative approach we took in segmentation of words into syntactic tokens.

The sentences in the treebank include all numbered examples in the grammar book. We have also included some in-text examples. Sentences with optional words or phrases are repeated with all alternatives suggested by the example. If a sentence has multiple, ambiguous interpretations listed in the grammar book, the sentence is repeated and annotated for each alternative analysis (2 sentences with four analyses, 2 with three analyses and 28 with two analyses).

All words are analyzed using TRmorph [7] and disambiguated with a simple morphological analyzer [8]. Morphological analyses are checked and corrected manually. The tokenized and morphologically analyzed sentences were annotated following current specifications of UD, using BRAT [18]. During this process, features or constructions that are not covered by the UD specifications are noted, and treebank-specific annotation guidelines are developed.

All sentences in the treebank are annotated by a single annotator (the author). Pending approval of publisher of the grammar book, we intend to release the treebank (the source sentences and the annotations) with a free/open-source license.

3 Issues in dependency annotation of Turkish

This section discusses some of the major annotation decisions. We focus mainly on the issues that conflict with the current UD specification. Most of these issues relate to morphological complexity of the language. All annotation decisions reflecting the current state of the treebank are documented separately, and it will be proposed as the Turkish-specific UD guidelines after the major issues are resolved.

3.1 Sub-word syntactic units

Turkish exhibits a highly productive derivational morphology. In some cases, the derivational suffixes may be attached late in the affixation process, causing an already inflected word to change its part of speech. This may result in conflicting feature-value assignments within the same word, and parts of a word may participate in different syntactic relations. As a result, taking words as syntactic tokens produces less than satisfactory syntactic analyses of Turkish sentences. The last word in (1) demonstrates a case where both of these problems are present.

(1) *Kaygımız* *terörün* *durdurulamamasıydı*
Worry.P3PL terror-GEN stop.CAU.PASS.ABIL.NEG-INF.P3S-COP.PAST.3S
'Our worry was (the fact that) terror could not be stopped.'

The word *durdurulamamasıydı* starts with the verb *dur* 'stop', inflects for *passive* and *causative* voice. The morpheme coded as 'ABIL' modifies the *mood* of the verb, and the verb is also negated. Next, this inflected verb is nominalized by a subordinating suffix and inflected for third person singular possessive agreement.[1] At this point, the clause can approximately be translated to English as 'the fact/case/event that (it/something) cannot be stopped'. Finally, the resulting noun is again verbalized through a copular suffix which carries the third person singular agreement.

The morphological complexity presented in the example above causes both of the problems mentioned above:

1. The same word may contain conflicting lexical/morphological features. For example, in (1) above, although the content verb *dur* is negative, the predicate introduced by the copula is positive.

2. Parts of the word may participate in different, conflicting, syntactic relations. In the example above, the subject of the verb *dur* 'stop' is *terör* 'terror', while the subject of the copular predicate is *kaygımız* 'our worry' (see Figure 1 for the dependency analysis).

These two issues arise with numerous other constructions in the language. We will revisit some of them in this paper.

The solution used for this problem in Turkish NLP literature is to split the words into multiple syntactic tokens, commonly referred to as *inflectional groups* (IG) [11]. In earlier Turkish NLP work, e.g., in METU-Sabancı treebank, words are split at all productive derivational suffixes. Many other suffixes, including the voice and modality suffixes discussed above, also introduce new IGs. For example, the word *durdurulamamasıydı* would be split into six IGs in the METU-Sabancı treebank (*dur-dur-ul-ama-ması-ydı* as opposed to *durdurulama-ması-ydı* in our annotation scheme). We introduce new IGs more conservatively: a word is split into multiple IGs only if (i) the parts of the word may carry the same feature and/or (ii) the parts may participate in different syntactic relations. Following these principles, we explicitly define the morphological contexts in which a new IG is introduced.

Another fundamental difference of our work and the METU-Sabancı annotation scheme is the annotations of relations between the IGs in a word with multiple IGs. The original version of the METU-Sabancı treebank does not specify the dependency relations between the IGs within a word explicitly. The last IG is always assumed to be the head of the other IGs within the word. No explicit or implicit structure is defined for relating the head and the dependent IGs. The version used during CoNLL-X shared task [5] introduces an explicit/dummy dependency label, `DERIV`, that relates the last IG (the head) to the other IGs in the word by a chain-like structure. In the present work, we always use dependency labels from UD dependency inventory to reflect the relations between the IGs. Furthermore,

[1] The suffix here in fact does not mark for possession, but indicates the subject of the verb.

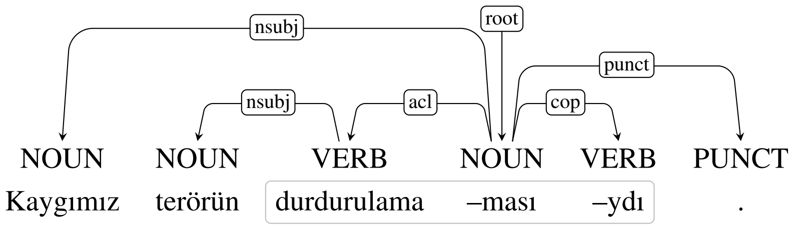


Figure 1: Analysis of (1), which includes a verbal noun. The details of the analysis are discussed in 3.5.

Token	Form	Lemma	UPOS	Feats	Head	Deprel
1	Kaygımız	kaygı	NOUN	Number=Sing|Number[psor]=Plur|Person[psor]=1	4	nmod
2	terörün	terör	NOUN	Case=Gen|Number=Sing	3	nsubj
3-5	durdurulamamasıydı	_	_	_	_	_
3	durdurulama	durmak	VERB	Mood=Abil|Negative=Neg|Person=3|Voice=Cau-Pass	4	acl
4	-ması	-me	NOUN	Number=Sing	0	root
5	-ydı	-0	VERB	Mood=Ind|Negative=Pos|Person=3|Tense=Past	4	cop
6	.	.	PUNCT	_	4	punct

Figure 2: The analysis of (1) in CoNLL-U format. All language specific columns (including the XPOS column which normally is the fifth column) and some features with default values (e.g., `Tense=Pres` from token 3, and `Number=Sing` from both predicate tokens) are left out for readability. The forms of the morphemes on column 2 are added for demonstration. Currently, the forms of the suffixes are left unspecified (annotated as '_').

following the UD preference for marking the content words as heads, we do not always mark the last IG as the head of the other IGs in the word.

Figure 1 demonstrates the dependency analysis of the example sentence in (1) graphically, and Figure 2 presents the same analysis in CoNLL-U format. Since some suffixes are altered (and sometimes deleted) based on morpho-phonological context, determining surface forms of IGs is sometimes non-trivial. Current version of the treebank leaves the surface forms for non-root IGs unspecified. The lemma field is always filled consistently for both root and non-root IGs.

3.2 Morphological features

The morphological complexity of the language requires special attention to the morphological features assigned to each syntactic unit. Many linguistic functions that are expressed through word order or function words in English are expressed using inflectional suffixes in Turkish. For example, a verbal root may receive over 10 inflectional suffixes, some of which may repeat multiple times. All IGs in the treebank are annotated with the lexical and inflectional features. We used features from the UD feature inventory as much as possible, and introduced new feature labels and/or values when necessary. Table 1 lists the features and their values.

Table 1: The features used in the treebank. The features or values not in the current UD specification are *emphasized*. For definitions of the existing features and values, the reader is referred to UD specification at `http://universaldependencies.github.io/docs/`.

Feature	Possible values	POS
Aspect	Perf, Prog, *Hab*, *Rapid*, *Dur*, Pro	VERB
Case	Acc, Dat, Gen, Ins, Loc, Nom	NOUN, PRON, PROP
Definite	Def, Ind	DET
Degree	Cmp, Sup	ADV
Evidential	*Fh*, *Nfh*	VERB
Mood	*Abil*, Cnd, Des, *Gen*, Imp, Ind, Nec, *Prs*	VERB
Negative	Neg, Pos	VERB
Number	Plur, Sing	NOUN, PRON, PROP, VERB
Number[psor]	Plur, Sing	NOUN, PRON, PROP
NumType	Card, Dist, Ord	NUM
Person	1, 2, 3	NOUN, PRON, PROP, VERB
Person[psor]	1, 2, 3	NOUN, PRON, PROP
PronType	Dem, Int, Loc, Prs	PRON
Reflex	Yes	PRON
Tense	Fut, Past, Pres, Pqp	VERB
VerbForm	Part, Trans	VERB
Voice	Cau, Pass, Rcp, *Rfl*	VERB

Here we will discuss the features and/or values that diverge from their traditional interpretation or from the current UD specification.

In Turkish, `Case` is an inflectional feature of nouns (POS tags `NOUN`, `PROPN` and `PRON`). Besides the five cases accepted in traditional grammars, we also use the case label `Ins` for instrumental or comitative marker *-(y)lA*.[2] We also use the same label when the suffix is not used in this case-like function but as a coordinating conjunction. The treatment of *-(y)lA* is similar to the METU-Sabancı treebank. Besides the suffix *-(y)lA*, there are a few productive suffixes (most notably *-lI* 'with', *-sIz* 'without') with case-like functions. Like the case-marked nouns, the derived word often functions like adverbs or adjectives. In this usage, it is possible to introduce non-standard case labels, or specific inflectional features for annotating these forms. However, we split these suffixes, and treat them like postpositions. The suffix is attached to the noun with the `case` relation. See Section 3.3 for more discussion on splitting productive suffixes.

The most challenging aspects of the inflectional features are related to verbal

[2] In describing variable suffixes we use capital letter 'A' to denote alternative letters 'e' or 'a', capital letter 'I' for 'ı', 'i', 'u', 'ü', capital letter 'C' is used for 'c' or 'ç'. Buffer consonants or vowels are written in parentheses. According to this notation, the forms *-(y)lA* can take based on the morpho-phonological context are *-la*, *-le*, *-yla* and *-yle*.

features. One aspect that currently does not fit well into the UD framework is the `Voice` feature. Turkish verbs can be inflected for reciprocal (`Rcp`), reflexive (`Rfl`), causative (`Cau`) and passive (`Pass`) voice. Current UD specification does not list `Rfl` as a possible voice value.[3] Additionally, current UD specification does not allow combination of voice values, e.g., for verbs that are inflected for both passive and causative voices as in (1) above, which occurs often in Turkish. A further complication is caused by the fact that the causative suffix is recursive. Even though it is very rare to see more than two iterations, a verb can be made causative multiple times, without a principled limit. For lack of an agreed solution, we currently annotate multiple `Voice` values as a list (see annotation of token 3 in Figure 2).

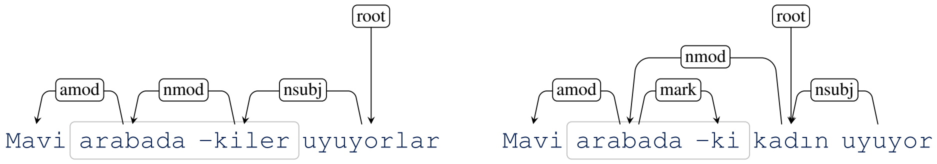
Despite the fact that the voice suffixes are considered as inflectional suffixes by descriptive grammars, METU-Sabancı treebank introduces a new IG for each voice feature. Since none of the IGs but the last one can be inflected, this creates 'inflectional groups' without any potential inflections. In other words, feature conflicts are not possible. The intermediate IGs cannot be modified by syntactic relations either.[4] As a result, the voice suffixes fail on both criteria set in Section 3.1 for introducing new syntactic tokens.

Turkish has a complex tense/aspect/modality (TAM) system. A single TAM suffix often marks a combination of tense, aspect and modality. Similar to [20], we annotate *evidentiality* as another feature dimension alongside tense, aspect and modality. We introduce a new feature, `Evidential` with two possible values `Nfh` (non-first hand) and `Fh` (first hand). We also use the following `Aspect` and `Mood` values that are not defined in the current UD specification.

- `Aspect=Hab` (habitual): *Güneş doğudan doğar* 'The sun rises from east'
- `Aspect=Dur` (durative): *bakakaldı* 'he/she looked (for a while, she was frozen while looking)' (durative stative) or *yapagelmiştir* 'he/she has gone on doing (something)' (durative progressive)
- `Aspect=Rapid` (for rapid or sudden action): *eve gidiver* 'quickly go home!'
- `Mood=Pers` (persuasive): *eve gitsene* 'go home (please)'
- `Mood=Abil` (abilitative or potentiality): *eve gidebilir* 'he/she may go home' or 'he/she is permitted to go home'. A negative verb may be inflected twice with this morpheme *eve gidemeyebilir* 'he/she may not be able to go home'

[3] The term *reflexive* here means that the subject of the predicate is also the direct object, i.e., the subjects acts on him/her/itself. This should not be confused with 'reflexive' verbs in some languages, e.g., German, which require an obligatory reflexive pronoun.

[4] One potential exception is that the subject of the non-causative predicate, i.e., content verb, may also be indicated by a noun phrase within the clause. In this case, the noun phrase acts as an argument or modifier of the complete (causative) predicate as well. Hence, we do not use another subject relation, but use language-specific subtypes of `dobj` and `nmod` relations (`dobj:cau` and `nmod:cau` respectively).



(a)

(b)

Figure 3: Dependency analyses of sentences in (2), demonstrating a nominal (a) and adjectival (b) derived with the suffix *-ki*.

- `Mood=Gen` (generalized modality): this marks statements with a more general or theoretical nature as opposed to statements of direct experience [10, p.295]. For example, *hastadır* ‘(I hypothesize/deduce that) she must be sick’ or *iki, iki daha dört eder* ‘two plus two is four’

Similar to voice suffixes, a verb may be inflected for multiple `Aspect` or `Mood` values. For example, *eve gidiverdim* ‘I went home (quickly)’ includes a completed (`Perf`) action that is performed quickly (`Rapid`). For multiple values, we follow the same strategy with multi-valued `Voice` features. Features in METU-Sabancı treebank are one-to-one mappings from the morphemes. As a result, a verb like *gitmiş* ‘he/she (evidently) left’ would be assigned a single `+Narr` (for narrative) feature. In our annotation scheme, the same verb receives tense, aspect, mood and evidentiality features `Tense=Past`, `Aspect=Perf`, `Evidentiality=Nfh` and `Mood=Ind`. Detailed documentation of these features and further examples can be found in the annotation guidelines document.

3.3 Productive derivational suffixes

As described in Section 3.1, some derivational suffixes cause an inflectional feature to be assigned multiple times, potentially with conflicting values. Example sentences in (2) demonstrate this with the suffix *-ki*. In (2a), the word *arabadakiler* refers to multiple people in the car. In the situation described, there are multiple people, but only a single car. Hence, *araba* ‘car’ carries the feature assignment `Number=Sing`, but *arabadakiler* ‘the ones in the car’ has the feature assignment `Number=Plu`. Furthermore, the adjective *mavi* ‘blue’ clearly refers to the car (not to the people), and the entity that is/are sleeping is the people, not the car. As a result, the suffix fulfils both criteria defined in Section 3.1 for introducing a new syntactic token.

(2) a. *Mavi arabadakiler uyuyorlar*
Blue car.LOC-ki.PL sleep.PROG.1P
‘The ones in the blue car are sleeping.’

b. *Mavi arabadaki kadın uyuyor*
Blue car.LOC-ki woman sleep.PROG.1S

‘The woman in the blue car is sleeping.’

If the suffix *-ki* derives an adjective as in (2b), admitting multiple units is not equally justified. We still observe that the adjective modifies *araba* ‘the car’, not the resulting adjective. This, however, is not unlike the case suffixes that often scope over the phrase headed by the noun they are attached to. A possible way to annotate the adverbial and adjectival forms could be introducing features for these suffixes. However, we currently split the word into multiple IGs in both uses of the suffix *-ki*.

Besides the suffix *-ki*, the suffixes *-lI*, *-sIz*, *-lIk*, *-sI* deriving (pro)nouns from adjectives and determiners and *-dIr* and *-lArI* that derive time adverbials introduce new syntactic units. In case the derivation results in an adjective or adverb, we mark the content word as the head, and attach the suffix using the dependency relation `case`. In case the derivation results in a noun, we mark the final (noun) IG as the head of the word. Figure 3 shows the dependency analyses for examples in (2). As a general rule, however, we do not split a derivational suffix if the word as a whole is lexicalized. For example, the word *kitaplık* (3a) is annotated as a single syntactic token, while it is annotated as two tokens in (3b).

(3) a. *Kitap*lık *dolu*
Bookshelf full
‘The *bookshelf* is full.’

b. *Çantamda* *üç* *kitap*lık *yer* *var*
Bag-P1S-LOC three book-lIk space exist
‘I have *space for* three *books* in my bag.’

3.4 Copular constructions and the null copula

The copular constructions in Turkish include the verb *ol-* ‘be / become’, the suffix *-(y)* attached to the subject complement or, with a much lower frequency, its clitic counterpart *i-*. We split the copular suffix and its inflections since the IG introduced by the copula carries features that conflict with the features of the subject complement. Figure 4 shows example analyses. In both analyses, the subject complement, *spor arabalar* ‘sports cars’, is plural. However, the in both examples the copula does not carry explicit inflections for `Number`, defaulting to the singular agreement. Furthermore, if the copular suffix is attached to a verbal noun, as shown in Figure 1, it may cause further feature conflicts. Whether they are suffixes, or free morphemes, copulas are always annotated as dependents (not as the head).

The analyses in Figure 4b shows a case where the copular suffix is not present in the sentence because of the morpho-phonological process. Since the suffix version of the copula is just a buffer consonant, with third person singular agreement combined with present tense, it is not realized on the surface. Although there is no overt copular suffix, the predicate in Figure 4b still carries the third person singular

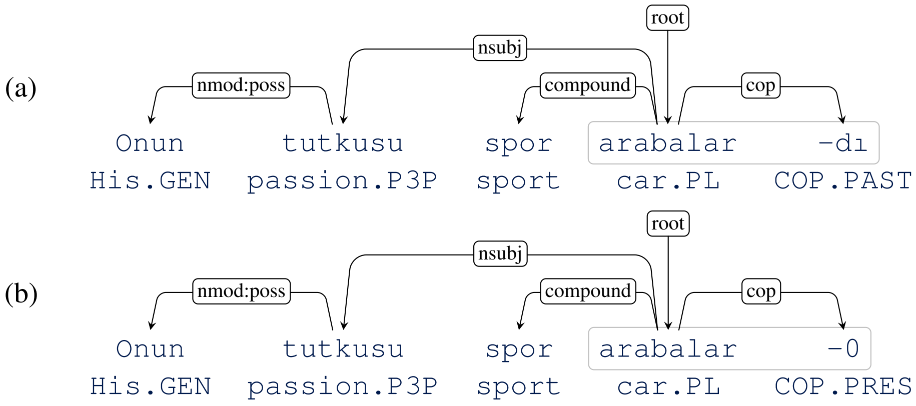
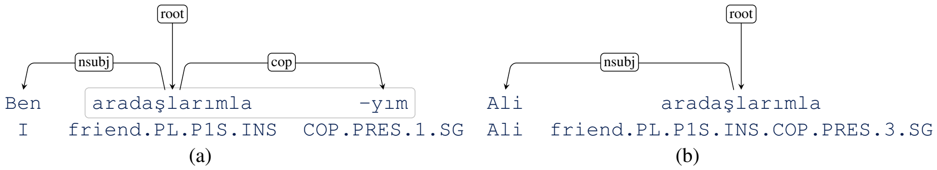


Figure 4: Example copula analyses (a) with overt past copula and (b) present copula with no overt suffix.



(a)

(b)

Figure 5: Inconsistent analyses of copula in case an empty syntactic unit is not introduced. (a) Overt copula: *Ben arkadaşlarımlayım* 'I am with my friends'. (b) No surface copula: *Ali arkadaşlarımla* 'Ali is with my friends'. Besides the conflicting number features (PL and SG) in (b), the same structure is analyzed differently.

agreement features, which conflicts with the plural number feature on the subject complement. As a result, we introduce an empty syntactic unit for the missing copula, despite UD's stand against null or missing elements. Besides the potential feature conflicts demonstrated above, failing to introduce the empty copular suffix results in analyses with different number of syntactic units for the same syntactic structure with trivial differences in their inflectional features. For example, the example sentences in Figure 5 differ only in the person agreement of the copular predicate. If we do not admit a null unit, as demonstrated in Figure 5, we assign different structures to these sentences.

The only exception where we do not introduce a null copula is in secondary predicates like *soğuk* 'cold' in *Ali çayını soğuk içer* 'Ali drinks his tea cold', or *arkadaş* 'friend' in *Ali'yi arkadaş sayarız* 'We consider Ali a friend'. The adjectives or nouns in these constructions are annotated with predicative relations without a copula.

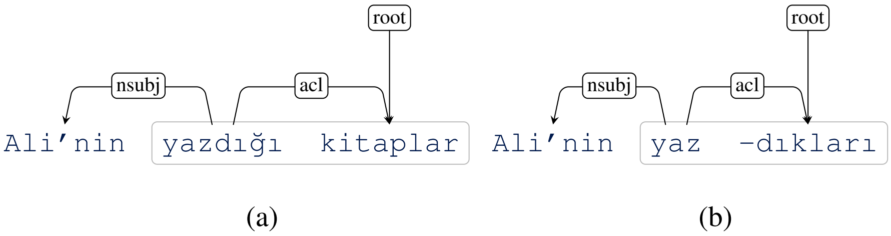


Figure 6: (a) A normal relative clause headed by a noun 'the books Ali has written'. (b) A headless relative clause 'the ones Ali has written'.

3.5 Non-finite subordinate clauses

The main means of subordination in Turkish is through a set of subordinating suffixes. Resulting subordinate clauses may function as *adjective*s, *adverb*s or *noun*s. Adjectival and adverbial constructions behave like the simple words with the same functions, and they do not receive further suffixes. As a result, we do not introduce a new IG in these cases, but assign a feature that indicates the verb form as *participle* and *converb* respectively [10, p.84].[5]

The verbal nouns, on the other hand, can be followed by most of the noun inflections. Furthermore, they can also be followed by POS-changing suffixes, most notably by the copular suffixes. An example of such a construction is given earlier in (1) and Figure 1. Figure 6 provides a simpler example with so-called *headless relative clauses* [10, p.389]. In this structure the head noun of a relative clause is omitted, and the relative clause is promoted to a (pro)noun referring to the missing noun phrase, and it can be inflected with all noun inflections. Note that in Figure 6b the predicate requires `Number=Sing`, while the resulting headless relative clause refers to multiple 'things', hence, having the feature assignment `Number=Plur`. Introducing a new syntactic token avoids this conflict. Although there are other conceivable solutions,[6] all other solutions would require major changes in the UD feature scheme. Besides solving potential feature conflicts, introducing a new IG makes the analysis similar to the 'headed' case shown in Figure 6a, and UD analysis of the corresponding English sentence where the pronoun 'one' would be analyzed as the head.

The conflict demonstrated in Figure 6 is very common for the headless relative clauses. With limited productivity, it also occurs with verbal nouns which denote entities of more abstract nature. This is demonstrated in (4) below, where the verb *kaç* 'run away' carries singular predicate-subject agreement feature, while the verbal noun *kaçmaları* formed by suffix *-mA* is plural.

[5] For converbs we use the label `Trans` since it has already been defined in the UD feature inventory, and the definition covers the converbs in Turkish.

[6] For example, by specifying all `Number` features as pertaining to *predicate* or the *noun phrase*.

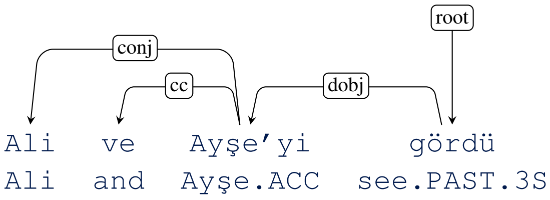


Figure 7: The analysis of sentence 'She/he saw Ali and Ayşe'. Note that annotating *Ali* as the head would make it difficult to search for accusative subjects, or mislead a parser to assign a subject relation rather than object, since the relevant feature is not immediately available on the head as it is in majority of the other cases. The problem becomes more severe when there are more than two conjuncts, and in case of covert coordination where no explicit conjunction or punctuation exists.

(4) *Ali'nin dersten kaçmaları annesini kaygılandırıyor*
Ali.GEN class.ABL run away-VN.PL.P3S mother.P3S.ACC worry.PROG.3S
'(The events of) Ali skipping classes worries his mother.'

3.6 Issues related to the dependency labels

Once the morphology of Turkish is represented well through the sub-word syntactic units and the additional features described above, annotating the syntax with existing UD dependency relationships is relatively straightforward. The only major divergence from the current UD scheme is related to the head direction in some of the constructions where the choice of head seems arbitrary (e.g., `conj` and `name`). For these relations, the UD specification requires a head-initial analysis. This results in suffixes that scope over the whole constituent to be attached to a non-head word, making it difficult to locate morphological features during a treebank search or during feature extraction for the statistical tools. Figure 7 presents an example. Currently, we annotate `conj` and `name` in a head-final fashion, otherwise following the UD guidelines where all the dependents are directly attached to the head.

Except the head-direction difference above, the only other noteworthy difference is additional dependency labels which are subtypes of the UD dependencies. Some of these subtypes are also used in other languages. Due to lack of space, we provide a list with brief descriptions. The reader is referred to the annotation guidelines for detailed descriptions of the dependency subtypes used. The additional dependencies currently in use are: `nmod:cau` and `dobj:cau` ('causee' of a causative predicate, see Section 3.2); `nmod:comp` (for comparatives); `nmod:pass` (actor of a passive predicate); `nmod:tmod` (temporal modifier); `nmod:own` (owner in a possessive existential sentence); `nmod:poss` (possessor in genitive-possessive construction); `nmod:part` (whole in a partitive construction); `compound:redup` (compounds formed by reduplication); `aux:q` (question particle).

4 Concluding remarks

This document introduced a Turkish grammar-book treebank following the UD annotation scheme. We believe that the current treebank could be a valuable resource for a number of purposes including (theoretical) linguistic research and testing NLP tools. We also see this effort as a first step towards constructing larger and better documented treebanks for Turkish that conform with the latest standards in dependency parsing and annotation.

Acknowledgments

The author gratefully acknowledges the financial support by the German Ministry for Education and Research (BMBF) as part of the CLARIN-D research infrastructure grant given to the University of Tübingen.

References

[1] Željko Agić, Maria Jesus Aranzabe, Aitziber Atutxa, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Jan Hajič, Anders Trærup Johannsen, Jenna Kanerva, Juha Kuokkala, Veronika Laippala, Alessandro Lenci, Krister Lindén, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Héctor Alonso Martínez, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Joakim Nivre, Hanna Nurmi, Petya Osenova, Slav Petrov, Jussi Piitulainen, Barbara Plank, Prokopis Prokopidis, Sampo Pyysalo, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Kiril Simov, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. Universal dependencies 1.1, 2015.

[2] Nart B. Atalay, Kemal Oflazer, and Bilge Say. The annotation process in the turkish treebank. In *Proceedings of the 4th International Workshop on Linguistically Interpreteted Corpora (LINC)*, 2003.

[3] Eva Pettersson Beáta Megyesi, Bengt Dahlqvist and Joakim Nivre. Swedish-turkish parallel treebank. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 470–473. European Language Resources Association (ELRA), 2008.

[4] Emily M Bender, Sumukh Ghodke, Timothy Baldwin, and Rebecca Dridan. From database to treebank: On enhancing hypertext grammars with grammar

engineering and treebank search. In *Electronic Grammaticography*. University of Hawai'i Press, 2012.

[5] Sabine Buchholz and Erwin Marsi. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164, 2006.

[6] Ruket Çakıcı. Wide-coverage parsing for turkish, 2008.

[7] Çağrı Çöltekin. A freely available morphological analyzer for Turkish. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 820–827, 2010.

[8] Çağrı Çöltekin. A set of open source tools for turkish natural language processing. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA), 2014.

[9] Çağrı Çöltekin. Turkish nlp web services in the weblicht environment. In *Proceedings of the CLARIN Annual Conference*, 2015.

[10] Aslı Göksel and Celia Kerslake. *Turkish: A Comprehensive Grammar*. London: Routledge, 2005.

[11] Dilek Z. Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36(4):381–410, 2002.

[12] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[13] Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. Building a Turkish treebank. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, chapter 15, pages 261–277. 2003.

[14] Victoria Rosén, Koenraad De Smedt, Paul Meurer, and Helge Dyvik. An open infrastructure for advanced treebanking. In Jan Hajič, Koenraad De Smedt, Marko Tadić, and António Branco, editors, *META-RESEARCH Workshop on Advanced Treebanking at LREC2012, Istanbul*, pages 22–29, 2012.

[15] Bilge Say, Deniz Zeyrek, Kemal Oflazer, and Umut Özge. Development of a corpus and a treebank for present-day written Turkish. In *Proceedings of the Eleventh International Conference of Turkish Linguistics*, 2002.

[16] Wolfgang Seeker and Özlem Çetinoğlu. A graph-based lattice dependency parser for joint morphological segmentation and syntactic analysis. *Transactions of the Association for Computational Linguistics*, 3:359–373, 2015.

[17] Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. A gold standard dependency corpus for english. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, 2014.

[18] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, 2012.

[19] Umut Sulubacak and Gülsen Eryiğit. Representation of morphosyntactic units and coordination structures in the Turkish dependency treebank. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 129–134, 2013.

[20] John Sylak-Glassman, Christo Kirov, Matt Post, Roger Que, and David Yarowsky. A universal feature schema for rich morphological annotation and fine-grained cross-lingual part-of-speech tagging. In Cerstin Mahlow and Michael Piotrowski, editors, *Systems and Frameworks for Computational Morphology*, pages 72–93. Springer, 2015.

[21] Heike Telljohann, Erhard Hinrichs, Sandra Kübler, and Ra Kübler. The tüba-d/z treebank: Annotating german with a context-free backbone. In *In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 2229–2232, 2004.

[22] Francis M. Tyers and Jonathan Washington. Towards a free/open-source universal-dependency treebank for kazakh. In *3rd International Conference on Computer Processing in Turkic Languages (TURKLANG 2015)*, 2015.

[23] Atro Voutilainen and Krister Lindén. Specifying a linguistic representation with a grammar definition corpus. In *Proceedings of corpus linguistics 2011*, 2011.

[24] Olcay Taner Yıldız, Ercan Solak, Onur Görgün, and Razieh Ehsani. Constructing a turkish-english parallel treebank. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 112–117. Association for Computational Linguistics, 2014.

[25] Daniel Zeman, Ondřej Dušek, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zdeněk Žabokrtský, and Jan Hajič. HamleDT: Harmonized multi-language dependency treebank. *Language Resources and Evaluation*, 48(4):601–637, 2014.

A Poor Man's Morphology for German Transition-Based Dependency Parsing

Daniël de Kok

Seminar für Sprachwissenschaft
University of Tübingen
E-mail: daniel.de-kok@uni-tuebingen.de

Abstract

It has been shown that the performance of statistical dependency parsing of richly inflected languages can be improved by giving the parser access to fine-grained morphological analyses. In this paper we show that similar levels of accuracy can be obtained without explicit morphological analysis, by implicitly learning morphology while training the parser.

1 Motivation

Transition-based dependency parsers [9] have traditionally relied on tokens and part-of-speech tags (or abstractions thereof) to find the best transition given a parser state. Recent work has shown that higher levels of accuracy can be achieved on morphologically-rich languages by using morphological analyses of the input. For instance, Ambati et al. [1] show that the use of features related to case, tense, aspect, and modality improve accuracy substantially on Hindi. Similarly, Marton et al. [12] show that features such as definiteness, person, number, and gender are helpful in parsing Arabic.

Since there is a certain amount of interaction between morphology and syntax in morphologically-rich languages, Lee et al. [10] explore joint morphological disambiguation and dependency parsing using a graph-based dependency parser. Bohnet and Nivre [4] propose a joint model for part-of-speech tagging and transition-based dependency parsing. These works show that joint processing is possible and can avoid error propagation through a natural language processing pipeline.

Since such joint approaches start with the assumption that morphology should be part of the output, they rely on morphology annotations and explicit morphological features. We propose a model that from the user's perspective does not use morphology at all. Instead, we will build up morphological representations 'implicitly' to obtain the same performance as a parser trained with morphology.

In contrast to the recent work of Ballesteros et al. [2], our model uses morphological representations as an addition to word embeddings. Furthermore, rather than using an LSTM recurrent neural network, our model uses a simpler and faster feed-forward neural network.

We believe that learning such morphological features in an unsupervised manner has a couple of benefits: one does not need a treebank with detailed morphology annotations, no morphological analyzer is required, and the parser is not limited to information that is provided by an inventory of morphological features.

2 Parser architecture

The architecture of our parser is inspired by Chen and Manning [5]. Their parser uses the arc-standard transition system [15] and a feed-forward neural network to select the best transition given the current parsing state. The parser state, which consists of a buffer β of unprocessed tokens and a stack σ of tokens currently undergoing processing, is represented as a dense vector that is the concatenation of the embeddings of words, part-of-speech tags, and head relations in the relevant positions of β and σ. This vectorized representation of the parser state forms the input of the neural net. Their approach has several advantages over earlier proposals that used symbolic features and linear discriminative classifiers: using a non-linear activation function on the hidden layer allows the network to infer combinatory features; the use of word embeddings increases lexical coverage significantly; and it is generally faster because it avoids costly feature construction [3].

In our parser, any position on the stack (σ_0^{n-1}) or buffer (β_0^{n-1}) can be addressed. For each token on the stack or buffer, five embedding layers can be consulted: the actual token (TOKEN), its part-of-speech tag (TAG), its morphological analysis (MORPH), its morphological analysis where every morphological feature is encoded as a one-hot vector (MORPH-ONEHOT), and the relation of a token to its head (DEPREL) if available. Moreover, the indirections LDEP and RDEP can be used to address the *n*-th leftmost or rightmost dependent of a token.

Figure 1 shows the (simplified) topology of our network. The input of the network consists of the concatenation of embeddings of the form TOKEN($\cdot$), TAG($\cdot$), DEPREL($\cdot$), MORPH($\cdot$) and/or MORPH-ONEHOT($\cdot$). The input is fed to a hidden layer using the hyperbolic tangent (*tanh*) activation function.[1] Finally, the output of the hidden layer is fed to the output layer, which uses the *softmax* function to obtain a probability distribution over all possible transitions. A simplification made in Figure 1 is that in reality, there are more hidden layers to train the relation embeddings. We refer to De Kok [7] for more information about the method used to train these embeddings.

[1] We found that the cube activation function proposed by Chen and Manning [5] often results in non-convergence.

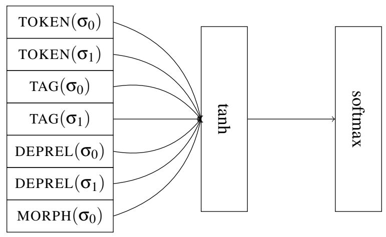


Figure 1: Simplified parser network topology, the number of inputs is reduced for illustrative purposes.

3 Use of a morphological analyzer

In a language processing pipeline where a morphological analyzer is used, analyses can be added as inputs to this parser in two different ways. For instance, suppose that the word at σ_1 is a singular and feminine, the morphological analyzer could assign a complex tag such as:[2]

`number:singular|gender:feminine`

The MORPH(σ_1) layer then simply returns the embedding of that morphological tag, which is derived from a large corpus (see Section 5.3). In contrast, the MORPH-ONEHOT(σ_1) layer constructs a feature vector. It decomposes the tag into individual features and represents each feature using a one-hot vector (or a zero vector if the feature is not relevant to a word class). The layer will then output the concatenation of the feature vectors as a representation of all the morphological features. Figure 2 shows the vector for the morphological tag above.

number		gender			tense	
1	0	1	0	0	0	0
singular	plural	feminine	masculine	neuter	present	past

Figure 2: One-hot vector encodings of feature-value pairs for the tag `number:singular|gender:feminine`

[2] For clarity, this example will only use three features.

4 Integrating morphology

As discussed in Section 1 our goal is to integrate morphological analysis in the parser, rather than using information from a morphological analyzer earlier in the pipeline. Our approach has four key points that we will discuss in this section, before showing how morphological analysis fits into our parsing network: (1) characters are represented as (pre-trained) embeddings; (2) words are morphologically represented as the concatenation of prefix and suffix embeddings; (3) for each word that is represented morphologically, a hidden layer is used for feature formation; and (4) the weights of all such hidden layers are tied.

Word representations Although tokens are already represented directly in the parser input using $\textsc{token}(\cdot)$, we add a new orthographical input representation $\textsc{char}(\cdot, p, s)$. This representation is the concatenation of the embeddings of the p prefix characters and s suffix characters of a particular token in the parser state. For instance, $\textsc{char}(\sigma_0, 2, 2)$ is the concatenation of the character embeddings of the prefix and suffix, both of length 2, of the token that is on top of the stack in the current parser state.

Character embeddings Our motivation for using pre-trained character embeddings, as opposed to e.g. one-hot encoding, is robustness. In case a character was not seen in the training data because it is not part of the German orthography, it could still be mapped closely to similar characters in vector space. To give a motivating example, the capitalized slashed o (*Ø*) does not occur in the training data. However, Danish names such as *Øresund* occur occasionally in German text. Since the suffix *-und* occurs in multiple word classes, knowing that the word begins with a capital biases the distribution towards singular nouns or proper nouns. In our character embeddings (see Section 5) the letter *Ø* is indeed clustered with other capital letters (Figure 3).

Feature formation Obviously, an affix of a particular length is not necessarily a linguistically meaningful affix. Our goal is to let the network learn what prefixes or suffixes are meaningful in the context of dependency parsing. To achieve this goal, each input of the form $\textsc{char}(\cdot, p, s)$ is fed through a hidden layer that uses the logistic function $g(x) = \frac{1}{1+e^{-x}}$ as the activation function.

Weight tying When $\textsc{char}(\cdot, p, s)$ inputs are extracted for multiple positions in the parser state, as is typically the case, each position obtains its own hidden layer. However, the weights for all such hidden layers are tied. If one hidden layer was used or multiple hidden layers with untied weights, the morphological analyses will differ per parser state position, even if the prefixes and suffixes are exactly the same. The outputs of these hidden morphology layers are then added as additional inputs to the hidden layer discussed in Section 2. We show the topology of the

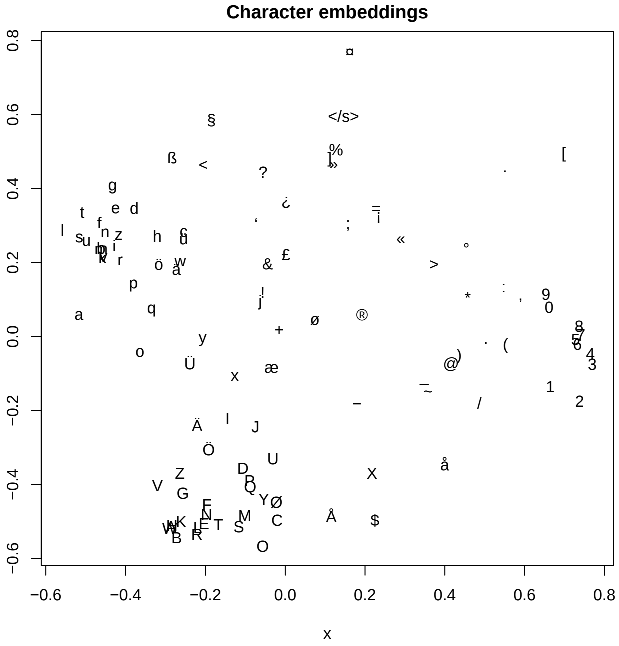


Figure 3: 2-dimensional MDS plot of the character embeddings trained on the TüPP-D/Z corpus. The capitalized slashed o (*Ø*) is clustered with other capitals.

network that integrates morphology in Figure 4.

The role of the hidden morphological layers can be interpreted in two related ways: (1) as extractors of morphological features that can be used in succeeding layers; or (2) as devices that can be used to create word embeddings such that morphosyntactically similar words are closer in vector space than dissimilar words.

5 Experimental setup

To evaluate the model that we propose, we compare a parser with this implicitly learned morphology (*morph-implicit*) to three parsers with and without access to analyses of a morphological analyzer. We will first describe the morphological analyzer used in our experiment, then we will give a description of the four parsers used in the evaluation. Since the parsers rely heavily on embeddings, we will then describe how the embeddings were trained. Finally, we will give a description of the training and evaluation procedure.

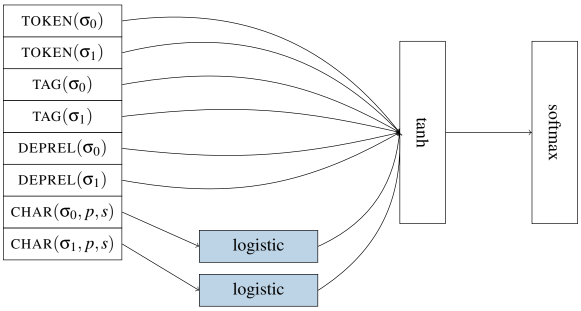


Figure 4: Feed-forward network that implicitly learns morphological features. The concatenated character embeddings of a word (CHAR$(\cdot, p, s)$) form the input of a hidden layer. The weights of such hidden morphology layers (here in blue) are tied.

5.1 Morphological analysis

The morphological analyses in our experiments are provided by RFTagger [18]. RFTagger is an HMM tagger that is tailored to tag sets with a large number of fine-grained tags. To avoid data sparsity, RFTagger splits complex morphological tags into attribute vectors and decomposes the context probabilities into products of attribute probabilities. This decomposition also allows for Markov models of a higher order, making it easier to capture long-distance dependencies. Finally, RFTagger can use a supplemental lexicon for improving coverage of words or word-tag combinations that are not seen in the training data.

For our experiments, we used the model and lexicon that was provided with RFTagger. The output contains the features *category*, *gender*, *case*, *number*, *grade*, *person*, *tense*, *mood*, and *finiteness*.

5.2 Parsers

no-morph The base parser does not have access to morphological information outside the relatively shallow analyses provided by part-of-speech tags. The parser uses pseudo-projective parsing [17] using the stack-projective transition system [16]. The neural network in this parser uses the concatenation of the following embeddings as its input: TOKEN(σ_0^3), TOKEN(β_0^2), TAG(σ_0^3), TAG(β_0^2), TOKEN(LDEP(σ_0^1)), TOKEN(RDEP(σ_0^1)), TAG(LDEP(σ_0^1)), TAG(RDEP(σ_0^1)), DEPREL(σ_0), DEPREL(LDEP(σ_0^1)), and DEPREL(RDEP(σ_0^1)).

morph This parser uses the same configuration as the *no-morph* parser, but adds the analyses of RFTagger to the input using the MORPH-ONEHOT layer, encoding morphological features using a sparse feature vector. The inputs MORPH-ONEHOT(σ_0^1) and MORPH-ONEHOT(β_0) are used in addition to the *no-morph* inputs.

morph-embed Like the *morph* parser, the *morph-embed* parser uses the output of a morphological analyzer. However, in the *morph-embed* parser, embeddings of complex tags are used via the MORPH layer. The inputs $\text{MORPH}(\sigma_0^1)$ and $\text{MORPH}(\beta_0)$ are used for morphology.

morph-implicit This parser also uses the transition system and inputs of the *no-morph* parser, but uses the method for learning morphological features that was described in Section 4. The following embeddings are added to the input vector: $\text{CHAR}(\sigma_0^1,4,4)$, $\text{CHAR}(\beta_0,4,4)$. We found empirically that using prefix and suffix sizes of 4 provides the best performance on our validation data.

5.3 Embeddings

The token and tag embeddings that are used in all four parsers and the morphology embeddings that are used in the *morph-embed* parser are trained on the TüBa-D/W [6] and TüPP-D/Z [14] corpora. The TüBa-D/W corpus contains 615 million tokens of German Wikipedia text and provides the necessary part-of-speech and morphological annotations. The TüPP-D/Z contains 204 million tokens from the German newspaper *taz*. From TüPP-D/Z we removed the articles that are overlapping with TüBa-D/Z [19]. Moreover, we reprocessed the corpus using the pipeline described by De Kok [6], so that the same annotation scheme is used as in the TüBa-D/W corpus.

The character embeddings for the *morph-implicit* parser are trained by treating each token as a sentence and each character as a token. The character embeddings were trained on only the TüPP-D/Z data, since it is cleaner than the TüBa-D/W.

The embeddings are trained using Wang2Vec [11], which is a modification of Word2Vec [13] that uses a structured skip n-gram model. In contrast to the unstructured model of Word2Vec, this model takes the proximity of words in the window to the focus word into account to create embeddings that are more tailored towards syntax-oriented tasks. For the word, tag, and morphology embeddings, we use the parameters suggested by Ling et al. [11]. For the character embeddings, we use smaller vectors of size 20.

5.4 Training and evaluation

All parsers are trained and evaluated with the dependency version [20] of the TüBa-D/Z release 9 [19]. The treebank, consisting of 85358 sentences and 1569916 tokens, is split into five interleaved parts. Two parts are used for training and validation. The remaining three parts are held out and used for evaluation.

6 Results

Table 1 shows labeled attachments scores for the three parsers. As we can see, all three parsers that use morphology perform better than the *no-morph* parser (significant at $p < 0.0001$). The difference is quite large if we take into account that the word embeddings already give the parsers very high lexical coverage — 97.13% of the tokens and 75.73% of the types in the evaluation data are known. The *morph-implicit* model that we propose in this paper is not only competitive the models that use a morphological analyzer, it even outperform them slightly (significant at $p < 0.05$).

Parser	LAS
no-morph	89.08
morph	89.35
morph-embed	89.40
morph-implicit	89.49

Table 1: Labeled attachment scores for the parsers with/without morphology.

As expected expected, the forward passes of the neural network of the *morph-implicit* parser are more expensive, since it applies extra hidden layers for morphological analysis. Since we did not implement the precompute trick [8] yet, for which Chen and Manning [5] report an order of magnitude speedup, the performance of the parser is highly dependent on having a performant BLAS library and a CPU that provides wide SIMD instructions. For this reason, we list the running time of each parser relative to the parser that does not use morphology in Table 2.[3]

Parser	Running time	Running time (+ RFTagger)
no-morph	1.00	
morph	1.58	2.73
morph-embed	1.03	2.18
morph-implicit	2.42	

Table 2: Running times for the parsers. The parsers using morphology have similar performance when including overhead of the morphological analyzer.

The use of morphology embeddings (*morph-embed*) has virtually no overhead compared to the *morph* parser. Encoding the morphology features using a sparse vector is, however, is over 1.5 times slower. The reason is that the *morph-embed* parser simply copies the embeddings into the input vector, while the *morph* parser has to split the complex tag first. The performance of the *morph* parser could be improved by pre-computing the vectors for complex tags. The parser proposed in this work (*morph-implicit*) is nearly 2.5 times slower as a result of the extra hidden layer(s). However, this comparison does not provide the complete picture, since the *morph* and *morph-embed* require the output of a morphological analyzer. The

[3] For reference, the *no-morph* parser processes 315 sentences per second using Intel Math Kernel Library using 4 threads on an Intel Xeon E5-2650 v3 @ 2.30GHz.

second column lists the running times of these parsers including morphological analysis. As we can see, the overall running times of the morphological parsers are roughly in the same ballpark.

7 Conclusion

We proposed a model for implicit morphological analysis, which can compete with the use of a morphological analyzer. This opens up the possibility to make competitive parsers using treebanks without morphological annotations, morphological analyzers, or hand-constructed morphology features.

An open question is if or how the model could be adjusted for languages that rely on infix morphology. Another interesting question is if the morphology features constructed in the network could also be used for morphological analysis where morphological tags are part of the output.

References

[1] Bharat Ram Ambati, Samar Husain, Joakim Nivre, and Rajeev Sangal. On the role of morphosyntactic features in hindi dependency parsing. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 94–102. Association for Computational Linguistics, 2010.

[2] Miguel Ballesteros, Chris Dyer, and Noah A Smith. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 2015.

[3] Bernd Bohnet. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97. Association for Computational Linguistics, 2010.

[4] Bernd Bohnet and Joakim Nivre. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1455–1465, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[5] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 1, pages 740–750, 2014.

[6] Daniël de Kok. TüBa-D/W: a large dependency treebank for german. In *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)*, page 271, 2014.

[7] Daniël de Kok. Bootstrapping a neural net dependency parser for German using CLARIN resources. In *Proceedings of the CLARIN 2015 conference*, 2015.

[8] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1370–1380, 2014.

[9] Sandra Kübler, Ryan McDonald, and Joakim Nivre. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1):1–127, 2009.

[10] John Lee, Jason Naradowsky, and David A Smith. A discriminative model for joint morphological disambiguation and dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 885–894. Association for Computational Linguistics, 2011.

[11] Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. Two/too simple adaptations of word2vec for syntax problems. *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL), Denver, CO*, 2015.

[12] Yuval Marton, Nizar Habash, and Owen Rambow. Improving arabic dependency parsing with lexical and inflectional morphological features. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 13–21. Association for Computational Linguistics, 2010.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. 2013.

[14] Frank Henrik Müller. Stylebook for the tübingen partially parsed corpus of written German (TüPP-D/Z). In *Sonderforschungsbereich 441, Seminar für Sprachwissenschaft, Universität Tübingen*, volume 28, page 2006, 2004.

[15] Joakim Nivre. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57. Association for Computational Linguistics, 2004.

[16] Joakim Nivre. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 351–359, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[17] Joakim Nivre and Jens Nilsson. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 99–106. Association for Computational Linguistics, 2005.

[18] Helmut Schmid and Florian Laws. Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 777–784. Association for Computational Linguistics, 2008.

[19] Heike Telljohann, Erhard Hinrichs, and Sandra Kübler. The TüBa-D/Z treebank: Annotating German with a context-free backbone. In *In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004*. Citeseer, 2004.

[20] Yannick Versley. Parser evaluation across text types. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, 2005.

Czech-English Bilingual Valency Lexicon Online

Eva Fučíková, Jan Hajič, Jana Šindlerová and Zdeňka Urešová

Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University in Prague
E-mail: {fucikova|hajic|sindlerova|uresova}@ufal.mff.cuni.cz

Abstract

We describe CzEngVallex, a bilingual Czech–English valency lexicon which aligns verbal valency frames and their arguments. It is based on a parallel Czech-English corpus, the Prague Czech-English Dependency Treebank (PCEDT), where for each occurrence of a verb, a reference to the underlying Czech and English valency lexicons (PDT-Vallex and CzEngVallex, respectively) is recorded. The CzEngVallex then pairs the entries (verb senses) of the two lexicons, and allows for detailed studies of verb valency and argument structure in translation and also compare the approaches to valency in the two languages on the background of the same underlying theory, the Functional Generative Description. The CzEngVallex lexicon is now accessible online, and we will also describe here the search interface which makes certain complex queries possible, using the lexicon and accessing the associated examples of verb sense translations, as extracted from the PCEDT corpus.

1 The PCEDT parallel corpus and its lexicons

Valency, or verb argument structure, is an important phenomenon both in linguistic studies as well as in language technology applications, since the verb is considered the core of a clause in (almost) every natural language utterance. Various dictionaries have been built - from Propbank [13] to Framenet [1] as well as various valency lexicons exist for several languages, such as Walenty [16] for Polish, Verbalex [8] or Vallex [9] for Czech, Valence Lexicon for a Treebank of German [3] for German etc. However, there are no truly multilingual valency dictionaries linked to corpora.

The Prague Czech-English Dependency Treebank (PCEDT 2.0) [4] contains the WSJ part of the Penn Treebank [10] and its manual professional translation to Czech, annotated manually using the tectogrammatical representation [11], first used for the Prague Dependency Treebank 2.0 (PDT) [5]. The tectogrammatical representation is in turn based on the Functional Generative Description theory [17].

The PCEDT contains 866,246 English tokens and 953,187 Czech tokens, aligned manually sentence-by-sentence and automatically word-by-word. It is annotated on all three annotation layers of the PDT: morphological, analytical (surface dependency syntax) and tectogrammatical (syntactic-semantic). However, as opposed to the PDT which is annotated fully manually[1], the PCEDT has been annotated for structure and valency at the tectogrammatical representation layer manually, but for POS, morphology and surface syntax only automatically.[2] Both sides of the tectogrammatical representation have been enriched with valency annotation, using two valency lexicons: PDT-Vallex for Czech and EngVallex for English.

The PDT-Vallex [6, 19, 18] is a valency lexicon originally developed for the PDT annotation. It contains almost 12,000 verb frames for about 7,000 verbs, roughly corresponding to verb senses found during the annotation of the PDT and the PCEDT. For each frame, verb arguments are listed together with their obligatoriness and required morphosyntactic realization(s). Each occurrence of a verb in the PDT (and the Czech side of the PCEDT) is linked to one verb frame in the PDT-Vallex lexicon.

The EngVallex [2] has been created for the English side of the PCEDT annotation. It is a semi-manual conversion of the Propbank frame files [13] into the PDT style of capturing valency information in valency frames, as used for Czech. The correspondence of the original Propbank entries and valency frames in the EngVallex is not necessarily 1:1 - entries have been occasionally merged or split. It contains over 7,000 frames for 4,300 verbs.

2 The CzEngVallex lexicon

The CzEngVallex lexicon [20, 21] is a bilingual valency lexicon with explicit pairing of verb senses (corresponding to valency frames) and their arguments, built upon the PCEDT. It contains 20,835 frame pairs describing the way verbal valency is mapped between languages, in particular between Czech and English.[3]

The lexicon draws on the Functional Generative Description Valency Theory (FGDVT). In this dependency approach, valency is seen as the property of some lexical items - above all verbs - to select for certain complementations in order to form larger units of meaning (phrase, sentence etc.). The governing lexical unit then governs both the morphosyntactic properties of the dependent elements and their semantic interpretation (roles). The number and realization of the selected

[1] With the exception of certain lexical node attributes.

[2] The surface dependency syntax on the English side has been derived from the Penn Treebank constituent syntax annotation, using head percolation rules, and thus can be considered semi-manual as well.

[3] This lexicon has been built within the project called “A comparison of Czech and English verbal valency based on corpus material (theory and practice)”, for more information, see https://ufal.mff.cuni.cz/czengvallex and https://ufal.mff.cuni.cz/biblio/?section=grant&id=-5269651103966024613&mode=view.

dependent elements constituting the valency structure of the phrase (or sentence) can be represented by valency frames, which can be listed in valency dictionaries. The basics of the FGD approach to valency can be found, e.g., in [14] or [7].

The annotation interface for the CzEngVallex is an extension of the tree editor TrEd [12][4] environment. It allows displaying and annotating sentential tree structures annotated on multiple linguistic layers with a variety of tags using either the Prague Markup Language (PML) format[5] or the Treex format.[6] Treex (formerly TectoMT) [22, 15] is a development framework for general as well as specialized NLP tasks (such as machine translation) working with tectogrammatically annotated structures.

This lexicon is a valuable resource to be used both for linguistically oriented comparative research, as well as for an innovative use in various NLP tasks. Its electronic version[7] is available from the repository of the Center of linguistic research infrastructure LINDAT/CLARIN in XML format and it is also available using a specific access portal in a searchable version, as described in this paper.[8]

It should be noted that not all verbs from the PCEDT can be found in the CzEngVallex: some verbs have not been translated at all as verbs, and vice versa, and some verb-verb translations have been so structurally different that they have not been included in the CzEngVallex.

Some of these cases can be extracted by inspecting the data where comments have been added by the annotators, and others by simple technical means (finding verbs with no matching alignment, finding verbs aligned to nouns, adjectives, or other structurally divergent tree segments).

	Verb	Frame	PCEDT Tokens	
Language	types	types	verbs	aligned
English	3,288	4,967	130,514	86,573
Czech	4,192	6,776	118,189	85,606

Table 1: Alignment coverage statistics - CzEngVallex/PCEDT

According to [20], 66% of English verb tokens found in the corpus have been aligned and can be found in the CzEngVallex (for Czech verb occurrences, it is 72%). Also, due to the fact that CzEngVallex is restricted to the parallel corpus only, it also covers only about 2/3rd of the underlying valency lexicons, the PDT-Vallex and the EngVallex. Exacts statistics are given in Table 1 (taken from [20]).

While both the underlying lexicons build upon the tectogrammatical representation used for both sides of the Prague Czech-English Dependency Treebank - there are the same five core arguments (`ACT`or, `PAT`ient, `ADDR`essee, `EFF`ect and `ORIG`in, about 40 additional free modifications, which might become obligatory

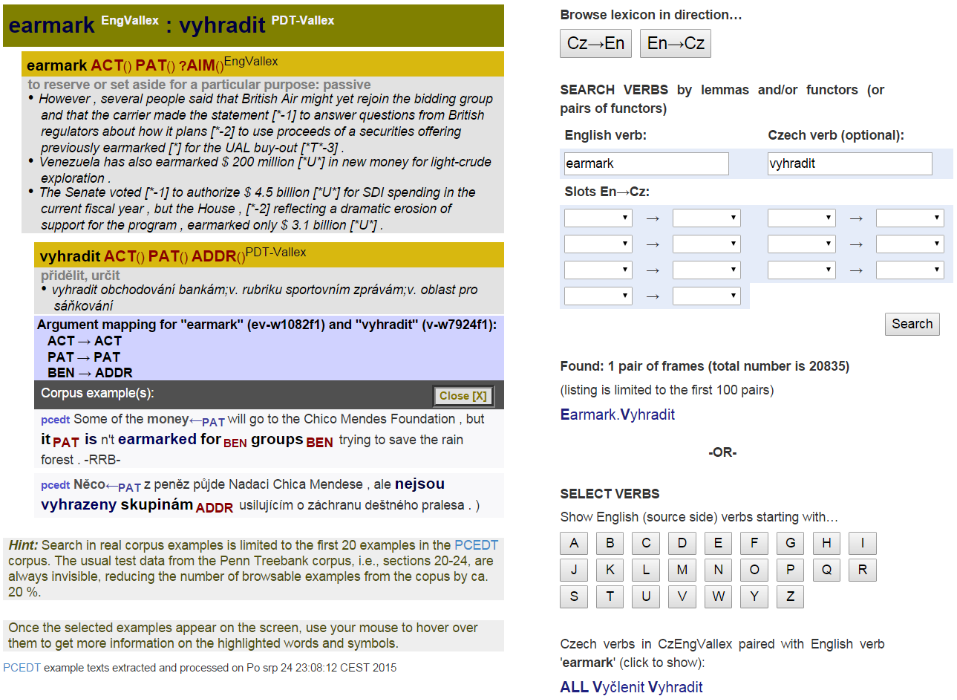
[4] http://ufal.mff.cuni.cz/tred

[5] http://ufal.mff.cuni.cz/jazz/PML

[6] http://ufal.mff.cuni.cz/treex

[7] http://hdl.handle.net/11234/1-1512

[8] http://lindat.mff.cuni.cz/services/CzEngVallex



earmark EngVallex : vyhradit PDT-Vallex

earmark ACT() PAT() ?AIM() EngVallex

to reserve or set aside for a particular purpose: passive

- *However , several people said that British Air might yet rejoin the bidding group and that the carrier made the statement [*-1] to answer questions from British regulators about how it plans [*-2] to use proceeds of a securities offering previously earmarked [*] for the UAL buy-out [*T*-3] .*
- *Venezuela has also earmarked $ 200 million [*U*] in new money for light-crude exploration .*
- *The Senate voted [*-1] to authorize $ 4.5 billion [*U*] for SDI spending in the current fiscal year , but the House , [*-2] reflecting a dramatic erosion of support for the program , earmarked only $ 3.1 billion [*U*] .*

vyhradit ACT() PAT() ADDR() PDT-Vallex

přidělit, určit

- *vyhradit obchodování bankám;v. rubriku sportovním zprávám;v. oblast pro sáňkování*

Argument mapping for "earmark" (ev-w1082f1) and "vyhradit" (v-w7924f1):
ACT → ACT
PAT → PAT
BEN → ADDR

Corpus example(s): Close [X]

pcedt Some of the money←PAT will go to the Chico Mendes Foundation , but it PAT is n't earmarked for BEN groups BEN trying to save the rain forest . -RRB-

pcedt Něco←PAT z peněz půjde Nadaci Chica Mendese , ale nejsou vyhrazeny skupinám ADDR usilujícím o záchranu deštného pralesa .)

Hint: Search in real corpus examples is limited to the first 20 examples in the PCEDT corpus. The usual test data from the Penn Treebank corpus, i.e., sections 20-24, are always invisible, reducing the number of browsable examples from the copus by ca. 20 %.

Once the selected examples appear on the screen, use your mouse to hover over them to get more information on the highlighted words and symbols.

PCEDT example texts extracted and processed on Po srp 24 23:08:12 CEST 2015

Browse lexicon in direction...

Cz→En En→Cz

SEARCH VERBS by lemmas and/or functors (or pairs of functors)

English verb: earmark

Czech verb (optional): vyhradit

Slots En→Cz:

Search

Found: 1 pair of frames (total number is 20835)
(listing is limited to the first 100 pairs)

Earmark.Vyhradit

-OR-

SELECT VERBS

Show English (source side) verbs starting with...

A B C D E F G H I
J K L M N O P Q R
S T U V W Y Z

Czech verbs in CzEngVallex paired with English verb 'earmark' (click to show):

ALL Vyčlenit Vyhradit

Figure 1: Main browsing and search form with search result for "earmark" → "vyhradit"

for any given verb, etc. - they also inevitably differ in several respects. First, instead of writing notes and examples to distinguish between verb senses of the individual valency frames, the creators of EngVallex often left a non-obligatory free modification in the valency frame, especially if they also found it in PropBank (where there is no obligatory/non-obligatory distinction being made). Such a free modification thus might be sometimes surprising to someone working only with PDT-Vallex so far. Also, the interpretation of certain label definitions such as ADDR vs. BEN was sometimes slightly different, as well as the conventions for using PAT and EFF with "verba dicendi" (say, explain, write, ...), and also the treatment of idioms and light verbs. These differences often show in the results of the searches as described below, and they do not represent "true" translation differences, but rather a difference in the application of the FGDVT theory and the tectogrammatical annotation guidelines to the two languages. Nevertheless, we consider that a unique opportunity to discover and study these differences through CzEngVallex (and its online search interface).

3 Online searching and browsing

The CzEngVallex lexicon is available at `http://lindat.cz` among “More Apps”, as the PDT-Vallex and EngVallex lexicons are.[9] The main search interface asks for a source verb (the direction might be switched using the En→Cz and Cz→En buttons), and either one of possible translations[10] to show all translations found in the CzEngVallex lexicon. If one of the verb input fields is left empty, the list of all translations will be displayed, allowing to directly select only one pair for a full display.[11]

Once all possible translation pairs are displayed, clicking on one of them shows the linked valency frames and the argument mapping within them. Fig. 1 shows the screen for the source verb “earmark” linked to “vyhradit”, all translations of “earmark” (lower right part of the screenshot), and the linked valency frames in the left column, with the following color coding: olive color is used as the background for the verb pair, dark yellow for the frame headline (with red-coded argument labels in it), and light violet/blue for argument mapping. Comments and examples recorded directly in the lexicon are on a light grey background. Corpus examples are separated by a dark grey bar (they can be hidden or made visible by a single click).

The same color coding is used on the webpages of the two underlying monolingual lexicons, PDT-Vallex and EngVallex, which are accessible by direct links[12] from the CzEngVallex entries.[13] There are two types of links to these monolingual lexicons - the two links in the headline (with olive color background) lead to the complete entry, while the PDT-Vallex/EngVallex links at the individual valency frame pairs (color-coded dark yellow) lead directly to the particular PDT-Vallex and EngVallex valency frames, respectively. The monolingual lexicons can be used for getting more information, such as more corpus examples or the morphosyntactic information for individual frame slots; for many verbs in PDT-Vallex, there are also additional verb senses, namely those used in other corpora than the PCEDT.

If the user is not satisfied with the selected pair, “ALL” can be selected (see lower right of Fig. 1), at the beginning of the list of plain translation equivalents), and the user is presented with the verb *earmark* and its two possible translations (Fig. 2).

If the user clicks on the `Show corpus examples` button, examples from the

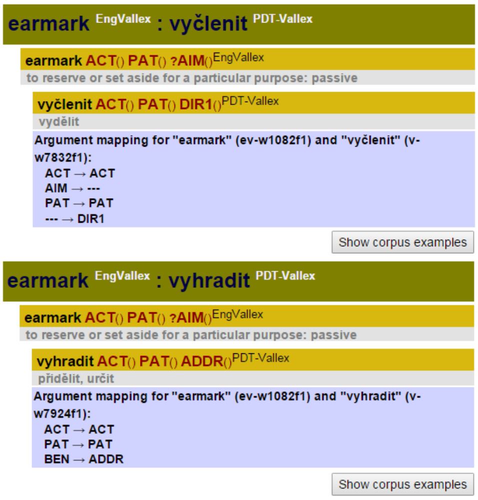
[9] Or directly at `http://lindat.mff.cuni.cz/services/CzEngVallex`.

[10] Or any of the fields (but not all) can be left blank.

[11] The verbs should be given in base form (as a lemma). It is also possible to use a standard regular expression on the verb lemma, e.g. `[a-d].*` for all verbs starting with a, b, c, or d. Full string match is assumed (i.e. the `'^'` and `'$'` characters for string start and end should not be used). Please note also that the list of verb pairs shown is limited to the first 100 pairs only.

[12] Shown as “superscripts” at the displayed verb lemmas and frames in the left column, cf. also Fig. 1.

[13] Except, of course, the underlying lexicons do not contain the argument pairing. They are also accessible at their own websites independently: `https://lindat.mff.cuni.cz/services/PDT-Vallex` and `https://lindat.mff.cuni.cz/services/EngVallex`.



earmark EngVallex : vyčlenit PDT-Vallex

earmark ACT() PAT() ?AIM() EngVallex

to reserve or set aside for a particular purpose: passive

vyčlenit ACT() PAT() DIR1() PDT-Vallex

vydělit

Argument mapping for "earmark" (ev-w1082f1) and "vyčlenit" (v-w7832f1):

ACT → ACT
AIM → ---
PAT → PAT
--- → DIR1

earmark EngVallex : vyhradit PDT-Vallex

earmark ACT() PAT() ?AIM() EngVallex

to reserve or set aside for a particular purpose: passive

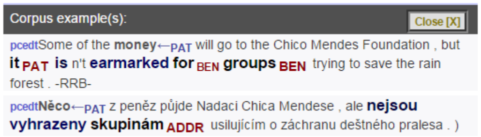
vyhradit ACT() PAT() ADDR() PDT-Vallex

přidělit, určit

Argument mapping for "earmark" (ev-w1082f1) and "vyhradit" (v-w7924f1):

ACT → ACT
PAT → PAT
BEN → ADDR

Figure 2: Two sense- and argument-aligned translations of *earmark* to Czech



Corpus example(s):

pcedtSome of the **money**←PAT will go to the Chico Mendes Foundation , but **it** PAT **is** n't **earmarked for** BEN **groups** BEN trying to save the rain forest . -RRB-

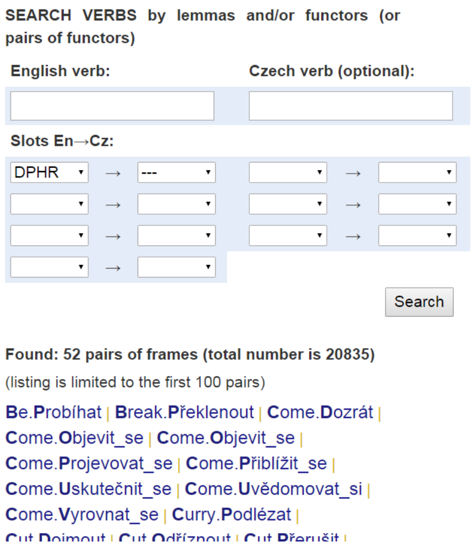
pcedt**Něco**←PAT z peněz půjde Nadaci Chica Mendese , ale **nejsou vyhrazeny skupinám** ADDR usilujícím o záchranu deštného pralesa .)

Figure 3: Corpus example for *earmark–vyhradit*

PCEDT parallel corpus are revealed (see Fig. 1 on the lower left, or as shown separately in Fig. 3). In the examples, presented as plain text, the verb and its arguments are highlighted based on the manual syntactic-semantic (tectogrammatical) annotation of the corpus. Arguments are also marked with the argument labels - in the example below, both PATs (including the one annotated only by co-reference) and the corresponding BEN- and ADDR-labelled arguments are shown. Both ACTors are elided in the passive construction used in the sentence and thus not shown, even if annotated in the tectogrammatical representation (and linked in the CzEngVallex).

It is also possible to search for particular argument types, specifying either side (Czech or English) or both, and moreover, any combination of argument pairs can be specified.[14] In every pair, either side can be left out (i.e. underspecified); it will then find all verb pairs where there is the entered argument on the specified side (language), and any argument on the other side to which it is linked. On the other hand, specifying a string of dashes `--` means that the particular argument must be marked as “not present” in CzEngVallex (same dashed string). For example, if a user wants to search for verb pairs where the English verb has the `DPHR` argument

[14] Up to 7, which is the maximum number of pairs found in the CzEngVallex entries.



Found: 52 pairs of frames (total number is 20835)

(listing is limited to the first 100 pairs)

Be.Probíhat | Break.Překlenout | Come.Dozrát | Come.Objevit_se | Come.Objevit_se | Come.Projevovat_se | Come.Přiblížit_se | Come.Uskutečnit_se | Come.Uvědomovat_si | Come.Vyrovnat_se | Curry.Podlézat | Cut.Dojmout | Cut.Odříznout | Cut.Přerušit |

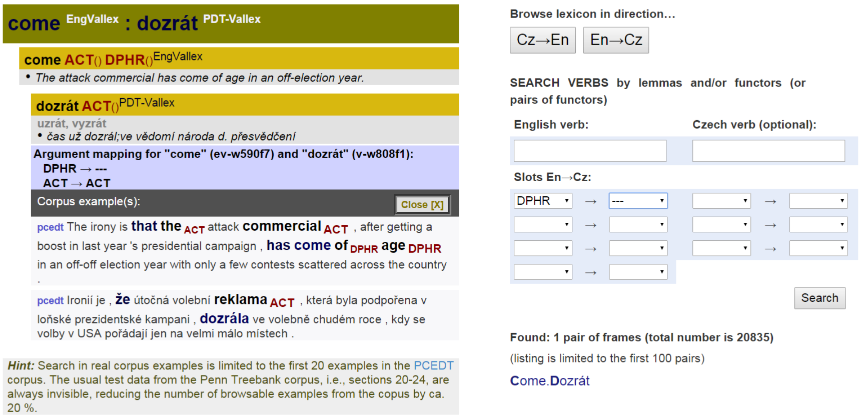
Figure 4: Intermediate result, searching for `DPHR`→`--`

while the Czech counterpart has an "empty" argument `--` linked to it, i.e., the English verb has a phrasal component while the Czech verb contains the phrasal meaning in the verb itself, these two arguments (in this order, provided that the direction En→Cz is selected) should be entered into a pair of "Slots" windows.

Once the search button is pressed, a list of all verb pairs that fulfill the conditions of the Slots query are displayed below the Search button (see lower part of Fig. 4). The user can visually check which pairs to display in full in the right-hand side of the screen, and then get them there by clicking on the particular pair. The list of pairs displayed might get very long, especially if a weak query (such as "show all pairs with ACT - ACT argument pairings") is entered; for that purpose, a count of pairs is displayed above the list to alert the user about the size of the list (and provide some statistics at the same time). In our case, if the user selects the "Come.Dozrát" pair, the resulting pair of frames and a corpus example is shown (Fig. 5).

Both the search by concrete verb (or a pair of verbs) and search by arguments can be combined. This is especially useful when searching within very frequent verbs with many verb senses (valency lexicon entries), such as `to be` or `to have`.

In addition, one can simply browse the lexicon using the letter-labelled buttons in the lower right part of the search interface (Fig. 1). After clicking on one of those buttons, a list of verbs starting the selected letter is displayed (can be long!), and a particular verb can be selected to see all possible senses of that verb and their pairings.



come EngVallex : dozrát PDT-Vallex

come ACT() DPHR() EngVallex

- *The attack commercial has come of age in an off-election year.*

dozrát ACT() PDT-Vallex

uzrát, vyzrát

- *čas už dozrál;ve vědomí národa d. přesvědčení*

Argument mapping for "come" (ev-w590f7) and "dozrát" (v-w808f1):
DPHR → ---
ACT → ACT

Corpus example(s):

pcedt The irony is that the ACT attack commercial ACT , after getting a boost in last year 's presidential campaign , has come of DPHR age DPHR in an off-off election year with only a few contests scattered across the country .

pcedt Ironií je , že útočná volební reklama ACT , která byla podpořena v loňské prezidentské kampani , dozrála ve volebně chudém roce , kdy se volby v USA pořádají jen na velmi málo místech .

Hint: Search in real corpus examples is limited to the first 20 examples in the PCEDT corpus. The usual test data from the Penn Treebank corpus, i.e., sections 20-24, are always invisible, reducing the number of browsable examples from the copus by ca. 20 %.

Browse lexicon in direction…

SEARCH VERBS by lemmas and/or functors (or pairs of functors)

English verb:

Czech verb (optional):

Slots En→Cz: DPHR → ---

Found: 1 pair of frames (total number is 20835)

(listing is limited to the first 100 pairs)

Come.Dozrát

Figure 5: Result for *come [of age]→dozrát*, example of DPHR→-- mapping

4 Conclusions

We have described some of the basic features of CzEngVallex, a bilingual valency dictionary created over the Prague Czech-English Dependency Treebank, a parallel corpus of 1 mil. words. The interlinked lexicons and the corpus are now publicly available online and searchable,[15] making it possible for a wide audience to get more insight into the use of verb arguments in translation, benefiting both in linguistic studies as well as in language technology, especially machine translation. The search interface is still under development, and new possibilities will be provided in future versions, such as search based on required morphosyntactic form of arguments, search within examples, “negative” search queries (for exclusions of certain pairings), etc. Moreover, we also consider updating the underlying corpora based on findings using the CzEngVallex, such as unifying the rules for argument labeling across languages; it would consequently improve the quality and consistency of CzEngVallex as well.

Acknowledgments

This work has been supported by the grant No. GP13-03351P of the Grant Agency of the Czech Republic and has been partially supported and has been using language resources developed, stored and distributed by the LINDAT/CLARIN project of the Ministry of Education of the Czech Republic (project LM2010013).

[15] And of course, for download: http://hdl.handle.net/11234/1-1512

References

[1] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[2] Silvie Cinková. From Propbank to Engvallex: Adapting the PropBank-Lexicon to the Valency Theory of the Functional Generative Description. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006), Genova, Italy*, 2006.

[3] Erhard Hinrichs and Heike Telljohann. Constructing a Valence Lexicon for a Treebank of German. In *Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories*, pages 41–52, 2009.

[4] J. Hajič, E. Hajičová, J. Panevová, P. Sgall, O. Bojar, S. Cinková, E. Fučíková, M. Mikulová, P. Pajas, J. Popelka, J. Semecký, J. Šindlerová, J. Štěpánek, J. Toman, Z. Urešová, and Z. Žabokrtský. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3153–3160, Istanbul, Turkey, 2012. European Language Resources Association.

[5] Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, Magda Ševčíková-Razímová, and Zdeňka Urešová. Prague Dependency Treebank 2.0, LDC Cat. No. LDC2006T01, also at http://hdl.handle.net/11858/00-097c-0000-0001-b098-5, 2006.

[6] Jan Hajič, Jarmila Panevová, Zdeňka Urešová, Alevtina Bémová, Veronika Kolářová, and Petr Pajas. PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation. In Erhard Nivre, Joakim//Hinrichs, editor, *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, volume 9 of *Mathematical Modeling in Physics, Engineering and Cognitive Sciences*, pages 57—68, Vaxjo, Sweden, 2003. Vaxjo University Press.

[7] E. Hajičová and P. Sgall. Dependency Syntax in Functional Generative Description. *Dependenz und Valenz–Dependency and Valency*, 1:570–592, 2003.

[8] Horák, Aleš and Pala, Karel and Hlaváčková, Dana. Preparing VerbaLex Printed Edition. In *Seventh Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2013*, pages 3–11, 2013.

[9] Markéta Lopatková, Zdeněk Žabokrtský, and Václava Ketnerová. *Valenční slovník českých sloves*. Karolinum, Praha, 2008.

[10] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *COMPUTATIONAL LINGUISTICS*, 19(2):313–330, 1993.

[11] Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, Zdeněk Žabokrtský, and Lucie Kučová. Anotace na tektogramatické rovině Pražského závislostního korpusu. Anotátorská příručka. Technical Report TR-2005-28, ÚFAL MFF UK, Prague, Prague, 2005.

[12] Petr Pajas and Peter Fabian. Tred 2.0 - newly refactored tree editor. `http://ufal.mff.cuni.cz/tred`, 2011.

[13] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106, 2005.

[14] Jarmila Panevová. On verbal frames in Functional generative description I. *Prague Bulletin of Mathematical Linguistics*, (22):3–40, 1974.

[15] Martin Popel and Zdeněk Žabokrtský. TectoMT: modular NLP framework. *Advances in Natural Language Processing*, pages 293–304, 2010.

[16] Adam Przepiórkowski, Elżbieta Hajnicz, Agnieszka Patejuk, Marcin Woliński, Filip Skwarski, and Marek Świdziński. Walenty: Towards a comprehensive valence dictionary of Polish. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, pages 2785–2792, Reykjavík, Iceland, 2014. ELRA.

[17] Petr Sgall, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Dordrecht, Reidel, and Prague, Academia, Prague, 1986.

[18] Zdeňka Urešová. *Valence sloves v Pražském závislostním korpusu*. Studies in Computational and Theoretical Linguistics. Ústav formální a aplikované lingvistiky, Praha, Czechia, 2011.

[19] Zdeňka Urešová. *Valenční slovník Pražského závislostního korpusu (PDT-Vallex)*. Studies in Computational and Theoretical Linguistics. Ústav formální a aplikované lingvistiky, Praha, Czechia, 2011.

[20] Zdeňka Urešová, Ondřej Dušek, Eva Fučíková, Jan Hajič, and Jana Šindlerová. Bilingual English-Czech Valency Lexicon Linked to a Parallel Corpus. In *Proceedings of the The 9th Linguistic Annotation Workshop (LAW*

IX 2015), Stroudsburg, PA, USA, 2015. Association for Computational Linguistics.

[21] Zdeňka Urešová, Eva Fučíková, and Jana Šindlerová. Czengvallex: Mapping valency between languages. Technical Report TR-2015-58, ÚFAL MFF UK, 2015.

[22] Zdeněk Žabokrtský. Treex – an open-source framework for natural language processing. In Markéta Lopatková, editor, *Information Technologies – Applications and Theory*, volume 788, pages 7–14, Košice, Slovakia, 2011. Univerzita Pavla Jozefa Šafárika v Košiciach.

Searching for Discriminative Metadata of Heterogenous Corpora

Gaël Guibon,[1] Isabelle Tellier,[1,2] Sophie Prévost,[1]
Matthieu Constant[3] and Kim Gerdes[2,4]

[1]Lattice CNRS, [2]Université Paris 3 – Sorbonne Nouvelle,
[3]Université Paris-Est, LIGM, [4]LPP CNRS
E-mail: gael.guibon@gmail.com,
isabelle.tellier@univ-paris3.fr, Matthieu.Constant@u-pem.fr,
sophie.prevost@ens.fr, kim@gerdes.fr

Abstract

In this paper, we use machine learning techniques for part-of-speech tagging and parsing to explore the specificities of a highly heterogeneous corpus. The corpus used is a treebank of Old French made of texts which differ with respect to several types of metadata: production date, form (verse/prose), domain, and dialect. We conduct experiments in order to determine which of these metadata are the most discriminative and to induce a general methodology.

1 Introduction

Labeled data used to train POS taggers or syntactic parsers by supervised machine learning technics are usually rather homogenous: The texts they come from share a common origin and most of their metadata. Yet, most actual text collections that NLP tools have to handle today are heterogenous in many ways. The consequence of this inadequacy is well known: Programs trained on homogenous texts by supervised machine learning approaches do not perform well when applied to new texts that differ from them in any important aspects, such as subject, genre or sublanguage. To address this problem, domain adaptation has become an important issue in machine learning NLP.

In this paper, we explore a very heterogenous corpus of Old French but the problem we tackle is not exactly domain adaptation. We want to use machine learning for *corpus exploration*, i.e. as a way to search for the most discriminative metadata of our texts. As a matter of fact, they belong to a *highly heterogenous treebank* and vary in dialect, domain, production date, and form (verse and prose). In this context, new questions arise: Which of these properties (metadata, in the following) influence most the language in which they are written? How can we best train a POS tagger and a parser on this treebank, in order to annotate a new yet unlabeled text of Old French? Is it better to use, as training data, a small homogenous corpus similar to the new text or a large heterogeneous dissimilar one? These questions can also be relevant for other heterogenous corpora and cross domain applications, so our purpose will also be to provide a general methodology.

2 Syntactic Reference Corpus of Medieval French

The SRCMF [1][11] is a treebank of Old French texts enriched with POS tags (chosen among 60 distinct possible ones) and fine-grained dependency structures (labeled with 31 distinct syntactic functions) which were built manually during an ANR-DFG joint research project (2009-2012). The corpus consists of 15 texts (245 000 words) whose electronic versions are stemming from the "Base de Francais Médiéval" (BFM)[2] [4] and the "Nouveau Corpus d'Amsterdam" (NCA)[3] [5]. The selection of the included texts is based on criteria such as date, form (verse/prose), domain (historical, literary...) and dialect. From the SRCMF we choose 10 texts, whose metadata are shown in Table 1.

Text	Date	Words	Form	Dialect	Domain
*Vie Saint **Légier***	late 10c.	1388	verse	n/a	religious
*Vie de Saint **Alexis***	1050	4804	verse	normand	religious
*Chanson de **Roland***	1100	28 766	verse	normand	literary
***Lapidaire** en prose*	Mid. 12c.	4708	prose	anglo-norm.	didactical
Yvain, Chr. de Troyes	1177-1181	41 305	verse	champenois	literary
*La **Conqueste** de Constantinople*, R. de Clari	>1205	33 534	prose	picard	historical
*Queste del Saint **Graal***	1220	40 417	prose	n/a	literary
***Aucassin** et Nicolete*	late 12c.-early 13c.	9844	verse & prose	picard	literary
Miracles from Gautier de **Coinci**	1218-1227	17 360	verse	picard	religious
*Roman de la **Rose*** from Jean de Meun	1269-1278	19 339	verse	n/a	didactical

Table 1: Texts of the Corpus and their Metadata

The syntactic annotation is based on a dependency model [12, 9], which distinguishes between, on the one hand, syntactic units and, on the other hand, different functions (such as "subject", "object", "adverbial", "auxiliary", "modifier"...), which specify the relation between a head and the structures depending on it.

[1] http://srcmf.org/

[2] http://bfm.ens-lyon.fr/

[3] http://www.uni-stuttgart.de/lingrom/stein/corpus/

3 Most Discriminative Metadata in SRCMF

In this section we propose a new experimental strategy to explore SRCMF based on metadata-related experiments, in order to find the metadata that are the most discriminative to tag and parse a new text.

3.1 Protocol of our Experiments

Our experiments are not designed to search for the best parser and configurations, this has already been studied before [8, 3]. But, to evaluate the impact of each metadata, it is important to obtain comparable results. So, we define a general protocol for all the experiments whose results are reported in this paper. For each metadata value, we process as follows:

- The data are split into a training set (of equal size for each possible value) and a test set made of the remaining content;
- On all training and test sets, the lemmas are predicted by *TreeTagger* trained on the *Nouveau Corpus d'Amsterdam* (NCA) parameters[4] [10];
- A CRF part-of speech tagger [6] is trained on the training set using *Wapiti* 1.4.0 [7] with the same templates taking into account the contexts of words and lemmas, word endings, etc. as in [3];
- A Dependency parser is trained on the training set (with Gold POS labels) with *Mate-tools* (anna-3.61 [5]) [1]
- The lemmatizer, the tagger and the parser are successively applied to the test set. The tagger is evaluated by its accuracy, the parser by the classical UAS and LAS values.

3.2 Summary of the Experiments

Dates. We first conduct these experiments using the production dates of the texts, which are usually considered as discriminative metadata. This approach seems all the more obvious for the SRCMF whose texts date from the late 11th to the 13th century, which allows us to make a comparison between centuries. Nonetheless, as the 11th century sub-corpus is too small (less than 7 000 words), we restrict ourselves to the other two. For this time slicing, we also have to remove *Aucassin* from the data, as it is not clear whether it has been written in the 12th or the 13th century.

The results show that applying a model trained on the same century gives higher scores, especially in LAS. In fact, even if high UAS, LAS, and accuracy

[4] `http://bfm.ens-lyon.fr/article.php3?id_article=324`

[5] Available at `https://code.google.com/p/mate-tools/downloads/detail?name=anna-3.61.jar&can=2&q=`

match with the correlation of century between train and test sets, the similarity between sub-corpora shows the contrary, with far more shared words when centuries differ, which is quite unexpected. This induces that the Out-Of-Vocabulary rate is not enough to predict the results. To confirm this intuition, we conducted few additionnal iterations of this experiment following the same method, in which texts from a century have been mixed, and a training set of 50 000 words[6] taken at random among them has been built. The performances reached by the models learned from these training sets lead to the same conclusions as previously. It is important to point out that even if both century based sub-corpora are composed of more than one domain and both verse and prose texts, the data from the 13th century are all written in Picard or in an undefined dialect. So, as one century does not vary in dialects, the date value is not independent from other metadata values. Moreover, as stated before, we could only use two centuries, which limits the scope of the results.

	Words	**Units**	**Sentence**	**Sentence length**
12th century [train]	50002	7337	5430	9
13th century [train]	50009	6934	3767	13
12th century [test]	24777	4844	2685	9
13th century [test]	60638	7597	4538	13

Table 2: Characteristics of the Time Sliced Sub-Corpora

Train \ Test		**12th century [test]**	**13th century [test]**
12th century	UAS	**88.81**	83.14
	LAS	**79.91**	71.93
	ACC	**94.69**	89.62
	Unknown | known words	**91.39** | 08.61	78.72 | **21.28**
	Different | shared lexicon	**61.20** | 38.80	28.59 | **71.41**
	Unknown | known words UAS	**81.05 | 90.00**	71.60 | 85.13
	Unknown | known words LAS	**66.42 | 81.47**	54.18 | 75.72
	Unknown | known words ACC	**87.29 | 95.39**	78.14 | 92.73
13th century	UAS	82.24	**89.07**
	LAS	69.24	**80.75**
	ACC	88.67	**94.62**
	Unkonwn | known words	73.83 | **26.17**	**92.25** | 07.75
	Different | shared lexicon	33.96 | **66.04**	**50.12** | 49.88
	Unkonwn | known words UAS	**76.94** | 86.61	74.35 | **88.77**
	Unkonwn | known words LAS	56.75 | 75.84	**57.96 | 80.46**
	Unkonwn | known words ACC	80.13 | 91.69	**85.31 | 95.41**

Table 3: Results Using Time Slicing

Forms. The impact of the production date has been clearly shown but other experiments have to be conducted to compare it with the effect of the other metadata.

[6] Because of the necessity to keep complete sentences, the size of randomly built training sets may vary by few words

We now consider text forms, i.e. if they are written in verse or prose. In the SRCMF, most of the texts are in verse but some of them are in prose and one text (*Aucassin et Nicolete*) is written in both forms. It is well aknowledged that verses are syntactically more constrained than prose, and moreover they have a richer lexicon (based on a higher amount of different units): We thus wonder if this could have an impact, in particular on parsing. This effect combines with the fact that sentences in prose are usually longer than those in verse. The results in Table 5 contradict the idea that a different text form induces a significantly different model of dependency parsing. Indeed, the prose test corpus obtains higher UAS and LAS even from the verse model. Compared to using centuries, this time shared lexicon and known words show a greater similarity between corpora of the same text form.

According to our results, text form seems to be useful to discriminate texts from a corpus but only when it comes to part-of-speech tagging, as dependency parsing results seem irrelevant to our purpose. This makes the form metadata useful, but less reliable than the time slicing, when it comes to corpus exploration.

	Words	**Units**	**Sentence**	**Sentence length**
prose [train]	41910	4320	4320	14
verse [train]	41907	6840	6840	7
prose [test]	36749	4370	4370	12
verse [test]	34478	4417	4417	10

Table 4: Characteristics of the Text Form Based Sub-Corpora

Train \ Test		**Prose [test]**	**Verse [test]**
Prose	UAS	**85.47**	76.33
	LAS	**74.96**	62.96
	ACC	**91.36**	83.61
	Unknown | known words	16.49 | **83.51**	**21.26** | 78.74
	Different | shared lexicon	57.02 | **42.98**	**77.05** | 22.95
	Unknown | known words UAS	**73.76 | 87.78**	65.87 | 79.15
	Unknown | known words LAS	**55.48 | 78.81**	46.37 | 67.44
	Unknown | known words ACC	**77.33 | 94.14**	76.78 | 85.46
Verse	UAS	**83.12**	82.79
	LAS	**71.52**	71.40
	ACC	90.06	**90.78**
	Unknown | known words	**18.81** | 81.19	14.03 | **85.97**
	Different | shared lexicon	**66,47** | 33,53	42.52 | **57.48**
	Unknown | known words UAS	**73.43 | 85.37**	72.39 | 84.49
	Unknown | known words LAS	55.45 | **75.24**	**55.62** | 73.98
	Unknown | known words ACC	81.02 | **92.15**	**84.13** | 91.86

Table 5: Verse [train] Shows Better Results on Prose [test] than on Verse [test], while Results on Known Words and Shared Lexicon Suggest the Contrary

Domains. Moving on to domain-related corpora, we now have more than two corpora to compare.

The notion of "domain" usually corresponds to what the texts are about. In the context of SRCMF, it is more related to the *literary genre* of these texts. Both notions do not exactly coincide but, in both cases, texts from the same domain should share some specific content words. As content words are less frequent and more ambiguous than grammatical words, we expect their presence or absence in both the training and test sets to affect the parsing results.

Domain adaptation is a prolific research field in machine learning and previous works have shown that it is possible to obtain better results by focusing on a specific domain rather than using a global approach [2]. This is why we could expect the domain value to have a great impact on the results.

A training set of about 16 000 words is first extracted from each of our four domain-specific sub-corpora, to ensure balanced training data. Test sets are made of the remaining content (Table 6). As expected, when the training and the test sets come from the same domain, the results, given in Table 7, are (in average) better and higher scores go along with the proportion of shared lexicon. This suggests that the domain is indeed a discriminative metadata. Moreover, results are quite stable except for the historical trainset, which shows an even greater gap when both sets come from the same domain, with an increase of about 30% in LAS. But, it is proper to remind that the historical corpora is made of only one text (*La Conqueste de Constantinople*).

	Words	**Units**	**Sentence**	**Sentence length**
Didactical [train]	16003	3820	1238	12
Historical [train]	16007	2298	1108	14
Literary [train]	16009	3529	1526	10
Religious [train]	16011	3645	1470	10
Didactical [test]	8013	2374	680	11
Historical [test]	17528	2414	1249	14
Literary [test]	104323	10828	10209	10
Religious [test]	7541	2182	708	10

Table 6: Domain-Specific Corpora's Charact.

Dialects. We then finally evaluate dialects as a discriminative metadata. A dialect speaker understands, at least partly, another dialect of the same language, as dialects share large parts of lexicon and grammar. In Table 9, we aim to determine whether or not the same holds for the dialects of our corpus. We use three sub-corpora based on the three distinct dialects, each training set being approximatively made up of 20 000 words.

We observe a huge increase in performance (of about 10 points in LAS, UAS, and accuracy) while applying a model on a same dialect. When the training and test sets do not stem from the same dialect, the shared lexicon is small. This could be due to the size of the SRCMF compared to contemporary language corpora, but more probably it is due to the heterogeneity of the texts in SRCMF, in particular concerning morpho-syntax and spelling, as shown in [3]. As a confirmation of the importance of shared lexicon: With an average sentence length of 7 words only,

Train \ Test	**Didactical[test]**	**Historical[test]**	**Literary[test]**	**Religious[test]**
Didactical [train]				
UAS	**81.78**	78.88	80.11	70.05
LAS	**71.23**	67.28	66.67	55.04
ACC	**90.75**	87.58	87.08	80.80
Unknown | known words	16.53 | **83.47**	**31.15** | 68.85	26.08 | 73.92	30.58 | 69.42
Different | shared lexicon	50.19 | **49.81**	78.05 | 21.95	**83.85** | 16.15	69.67 | 30.33
Unknown | known w. UAS	**71.68 | 83.78**	69.12 | 83.29	70.13 | 83.63	59.80 | 74.57
Unknown | known w. LAS	**53.93 | 74.66**	52.89 | 73.79	50.69 | 72.29	38.46 | 62.34
Unknown | known w. ACC	80.89 | **92.70**	**81.30** | 90.43	77.53 | 90.44	66.96 | 86.89
Historical [train]				
UAS	67.49	**90.07**	73.03	32.29
LAS	51.12	**82.20**	57.30	45.08
ACC	72.74	**95.66**	76.67	69.93
Unknown | known w.	41.09 | 58.91	08.08 | **91.92**	38.66 | 61.34	**42.57** | 57.43
Different | shared lexicon	81.94 | 18.06	46.67 | **53.33**	**90.46** | 09.54	79.84 | 20.16
Unknown | known w. UAS	58.08 | 74.05	**80.16 | 90.94**	65.06 | 78.05	52.80 | 69.33
Unknown | known w. LAS	38.24 | 60.11	**63.70 | 83.92**	45.20 | 64.93	31.56 | 55.10
Unknown | known w. ACC	62.67 | 79.77	**87.50 | 96.38**	66.95 | 82.80	57.20 | 79.38
Literary [train]				
UAS	77.22	82.02	**84.79**	73.09
LAS	64.07	70.79	**73.63**	59.01
ACC	85.10	88.95	**91.93**	83.25
Unknown | known w.	27.01 | 72.99	**27.35** | 72.65	14.42 | **85.58**	27.16 | 72.84
Different | shared lexicon	68.17 | 31.83	73.58 | 26.42	**75.36** | 24.64	65.96 | **34.04**
Unknown | known w. UAS	66.17 | 81.31	74.07 | 85.02	**74.28 | 86.56**	61.18 | 77.53
Unknown | known w. LAS	46.03 | 70.74	**57.21** | 75.90	56.25 | **76.55**	40.67 | 65.84
Unknown | known w. ACC	73.61 | 89.35	80.72 | 92.04	**82.50 | 93.51**	69.04 | 88.55
Religious [train]				
UAS	74.99	79.76	79.52	**80.72**
LAS	61.61	67.94	65.94	**69.35**
ACC	83.31	87.62	85.91	**90.16**
Unknown | known w.	29.01 | 70.99	**29.50** | 70.50	26.58 | 73.42	14.07 | **85.93**
Different | shared lexicon	71.66 | 28.34	76.56 | 23.44	**85.05** | 14.95	43.47 | **56.53**
Unknown | known w. UAS	63.98 | 79.48	**70.17 | 83.77**	69.16 | 83.28	68.61 | 82.70
Unknown | known w. LAS	44.10 | 68.76	**53.31 | 74.06**	49.00 | 72.07	49.58 | 72.59
Unknown | known w. ACC	71.73 | 88.05	**81.80** | 90.06	75.87 | 89.55	75.87 | **92.50**

Table 7: Experiments Using Domain Based corpora

the *Normand* corpus should be easier to parse than the other dialects. Results do not clearly exhibit such differences (except for UAS), probably because the rate of known words is lower when evaluated on the *Normand* test set.

In any case, dialect turns out to be the most discriminative metadata among those evaluated, when it comes to predicting parsing results on Old French.

To go even further with this metadata, we make a complementary “leave one out” experiment using the dialect-based corpus segmentation, in order to determine which of its three distinct values is the most different from a machine learning point of view. The results in Table 10 show that the *normand* dialect seems to be the most remote one with the lowest amount of shared lexicon and known words. It leads

	Words	**Units**	**Sentence**	**Sentence length**
Champenois [train]	20005	3283	1821	10
Normand [train]	20028	3799	2639	7
Picard [train]	20011	3565	1538	13
Champenois [test]	21301	3440	1970	10
Normand [test]	13542	5503	1784	7
Picard [test]	40727	3042	3205	12

Table 8: Dialectal Corpora Characteristics

Train \ Test	**Champenois[test]**	**Normand[test]**	**Picard[test]**
Champenois [train]			
UAS	**86.07**	78.61	76.66
LAS	**76.30**	61.93	63.63
ACC	**93.41**	81.17	84.02
Unknown/known words	10.23 | **89.77**	**51.05** | 48.95	31.20 | 68.80
Different/shared lexicon	51.09 | **48.91**	**82.09** | 17.91	79.56 | 20.44
Unknown/known words UAS	**73.83 | 87.46**	72.83 | 84.63	66.38 | 81.32
Unknown/known words LAS	**59.14 | 78.25**	51.34 | 72.98	46.29 | 71.49
Unknown/known words ACC	**84.57 | 94.41**	72.59 | 90.12	67.99 | 91.30
Normand [train]			
UAS	74.54	**88**	73.77
LAS	59.31	**77.96**	60.48
ACC	81.12	**93.31**	82.55
Unknown/known words	34.14 | 65.86	11.25 | **88.75**	**38.77** | 61.23
Different/shared lexicon	82.24 | 17.76	43.90 | **56.10**	**87.05** | 12.95
Unknown/known words UAS	64.37 | 79.81	**78.53 | 89.20**	64.19 | 79.84
Unknown/known words LAS	45.30 | 66.58	**60.21 | 80.21**	46.86 | 69.11
Unknown/known words ACC	72.54 | 85.57	**82.01 | 94.74**	72.50 | 88.92
Picard [train]			
UAS	77.35	79.41	**85.14**
LAS	63.46	63.20	**75.90**
ACC	84.40	82.11	**93.25**
Unknown/known words	24.58 | 75.42	**45.57** | 54.23	11.16 | **88.84**
Different/shared lexicon	74.51 | 25.49	**82.42** | 17.58	60.03 | **39.97**
Unknown/known words UAS	66.15 | 81.00	**72.60** | 85.24	71.49 | **86.86**
Unknown/known words LAS	47.03 | 68.81	51.47 | 72.98	**55.29 | 78.49**
Unknown/known words ACC	75.34 | 87.34	72.93 | 89.78	**80.56 | 94.85**

Table 9: Experiments Using the Dialectal Corpus Segmentation

to the lowest accuracy and LAS, however for both unknown and known words its UAS and LAS are higher than for the other two corpora. This can only be possible because it contains a higher rate of unknown words. This means that low proximity between corpora does not necessary brings lower unknown words recognition.

4 Towards a General Methodology

This exploration based on metadata opens interesting perspectives. Given a new text with its associated metadata as an input, one can expect to develop a gen-

The other 2 tested on	Champenois	Normand	Picard
UAS	84.18	**84.41**	82.06
LAS	**72.64**	70.41	70.98
ACC	**89.86**	86.56	88.54
unknown/known w.	17.04 | **82.96**	**37.45** | 62.55	25.05 | 74.95
different/shared lex.	63.08 | **36.92**	**75.97** | 24.03	71.55 | 28.45
unknown/known w. uas	75.25 | 86.01	**79.16 | 87.56**	73.56 | 84.91
unknown/known w. las	58.82 | 75.49	**59.84 | 76.74**	56.39 | 75.85
unknown/known w. acc	**84.29** | 91.01	79.66 | 90.69	76.97 | **92.41**

Table 10: Leave One Out experiments Using the Dialectal Corpus Segmentation

eral methodology to find the best tagging or parsing model. Let the associated metadata be a set of attribute-value pairs. For instance, assume that the new input text has the following set of attribute-value pairs: *century=13th*, *domain=literary*, *form=verse*, *dialect=picard*. From the previous experimental results, one can find the best tagging/parsing model among those available. The selected parsing model for the input text would be the one leading to the best LAS among those associated with the text metadata. In our example, we have four possible candidate models, each one associated with an attribute-value pair:

- century=13th (LAS=80.75 trained on *13th century [train]*)
- domain=literary (LAS=73.63, trained on *literary [train]*)
- form=verse (LAS=71.40 trained on *verse [train]*)
- dialect=picard (LAS=75.90, trained on *picard [train]*)

In this example, the best parsing model for *century=13th* seems to be the one trained on *13th century [train]*, reaching a LAS of 80.75 (cf. Table 7).

This proposed methodology is still rudimentary. We are aware that, in our corpus, the metadata are correlated and interfere with each other. For instance, in SRCMF, texts written in *Picard* are all from the 13th century. This cannot be avoided due to the lack of available texts, let alone tagged corpora, of Old French. Furthermore, there is an unavoidable part of an arbitrary in the way metadata values are defined (the various distinguished domains, the time slicing, knowing that changes do not occurs specifically at the turn of two centuries...). The different sizes of the training sets are also another issue for such a general method.

5 Conclusion

In this paper, we have shown that machine learning could serve as a very effective corpus exploration strategy. Each experiment helps us to better understand the specificities of our highly heterogenous corpus. The originality of the approach we have followed is that it is focused on *metadata discrimination*. Machine learning engineering is usually more concerned with feature selection or parameter optimization, applied to stable training and test sets to gain better overall results. Here, the machine learning devices used are stable, but we vary the way training and test corpora are built, in order to evaluate their influence on the final result.

This work can be extended in various ways. With SRCMF, the metadata selection we have started out with should be investigated further, with the goal to build a complete decision tree for a given new text whose metadata are known. The ideal decision tree would provide the best possible labeled sub-corpus to use as a training set, to build the best possible model for this given text. To achieve this goal, the correlations between distinct metadata should also be investigated further. We are prudent concerning the generalizability of some of our conclusions, because the impacts of metadata are mixed with other factors: size of the available sub-corpora, lexical variation, effect of the combination of metadata... More experiments are necessary to clarify each of them.

Nevertheless, we believe that this global approach could be relevant in many contexts, as heterogenous corpora are increasingly becoming an important subject of parsing technologies.

References

[1] Bernd Bohnet. Very high accuracy and fast dependency parsing is not a contradiction. In *The 23rd International Conference on Computational Linguistics (COLING 2010)*, Beijing, China, 2010.

[2] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520, 2011.

[3] Gaël Guibon, Isabelle Tellier, Matthieu Constant, Sophie Prévost, and Kim Gerdes. Parsing poorly standardized language dependency on old french. In *13th Treebank and Language Theory (TLT)*, 2014.

[4] Céline Guillot, Alexei Lavrentiev, and Christiane Marchello-Nizia. Document 2. les corpus de français médiéval : Etat des lieux et perspectives. *Revue française de linguistique appliquée*, XII:125–128, 2007.

[5] Pierre Kunstmann and Achim Stein. Le nouveau corpus d'amsterdam. In *"Actes de l'atelier de Lauterbad"*, pages 9–27, 2007.

[6] John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, pages 282–289, Seattle, Washington, 2001.

[7] Thomas Lavergne, Olivier Cappé, and François Yvon. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513, July 2010.

[8] Francesco Mambrini and Marco Passarotti. Will a parser overtake achilles? first experiments on parsing the ancient greek dependency treebank. In *Eleventh International Workshop on Treebanks and Linguistic Theories*, pages 133–144. Edições Colibri, 2012.

[9] Alain Polguère and al. *Dependency in linguistic description*, volume 111. John Benjamins Publishing, 2009.

[10] Achim Stein. Parsing heterogeneous corpora with a rich dependency grammar. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014. European Language Resources Association.

[11] Achim Stein and Sophie Prévost. Syntactic annotation of medieval texts: the syntactic reference corpus of medieval french (srcmf). In Tübingen: Narr, editor, *New Methods in Historical Corpus Linguistics*. 2013. Corpus Linguistics and International Perspectives on Language, CLIP Vol. 3, P. Bennett, M. Durrell, S. Scheible and R. Whitt (eds).

[12] Lucien Tesnière. *Eléments de syntaxe structurale*. Librairie C. Klincksieck, 1959.

Quantitative Evidence, Collostructional Analysis and Lexical Approaches to Argument Structure

Matías Guzmán Naranjo

Department of Linguistics
University of Leipzig
E-mail: matias.guzman_naranjo@uni-leipzig.de

Abstract

One of the most interesting recent debates within model theoretic, construction based, approaches to syntax has been that of lexical vs phrasal analysis of argument structure. I will present new quantitative evidence using a German Treebank that the patterns we find in language use are more compatible, and better explained by the lexical approaches than by the phrasal approaches. In doing so I present a new possible approach to collostructional analysis using treebanks.

1 Introduction

There is currently a debate within monostratal, constraint-based grammatical frameworks as to which is the best way of modelling argument structure and argument structure changes. On the one hand, Cognitive Construction Grammar (Goldberg [6, 9, 7, 8]) has developed a phrasal approach with argument structure constructions, and on the other hand HPSG and SBCG proponents (Sag et. al. [17], Ginzburg and Sag [5], Müller [13], Müller and Wechsler [14]) have strongly argued for a lexical, verb-based analysis.

In the phrasal approach individual verbs combine with phrasal constructions to license a given structure. A simple transitive like *John kills Mary* is licensed by the verb $\textit{kill}_{\text{(killer, killed)}}$[1] combining with a transitive constructions [$\text{SUBJ}_{\text{agent}}$ VERB $\text{DOBJ}_{\text{patient}}$][2]. In this approach the verb contains semantic information about thematic roles, while the syntactic properties and some additional meaning are provided by the passive construction. Valency alternations and valency changes are also handled by the use of argument structure constructions. This way, passives are

[1] The semantic roles assigned by the verb as written here in brackets besides the verb.

[2] Notice Cognitive Construction Grammar has no established formalization, and different authors choose very different notations.

the result of a verb combining with a passive construction like [SUBJ$_{\text{patient}}$ AUX VERB (PPO$_{\text{agent}}$)], resultatives are the product of a resultative construction, etc.

Lexical approaches take that most cases should be handled by lexical information on the verb. In these approaches there are no transitive constructions that license transitive sentences, instead, there are transitive verbs with a transitive **argument structure**: *kill*$^{trans-verb}$ [ARG-ST <NP$_{\text{agent}}$, NP$_{\text{patient}}$>][3]. Valency alternations are the product of unary lexical rules that take a verb of a given type with an argument structure and return a new verb with the same meaning but a different type and different argument structure. A passive rule, for example, would take a transitive verb *kill* as above and return a passive participle: *killed*$^{passive-part}$ [ARG-ST <NP$_{\text{patient}}$, (PP$_{\text{agent}}$)>]. Finally, general grammar rules enforce how these verb types combine with their arguments.

The recent debate on how to best account for argument structure patterns has developed using mostly qualitative evidence (Goldberg [8], Kay [11], Müller [13, 14]). The studies that have tried to argue from a corpus linguistics perspective have so far claimed to support the phrasal analysis of argument structure (Stefanowitsch and Gries [18], Gries and Stefanowitsch [10]).

The main quantitative argument for a phrasal approach comes from collostructional analysis (Stefanowitsch and Gries [18], Gries and Stefanowitsch [10]). The idea of collostructional analysis is that we can extend collocational analysis to grammatical patterns if we treat these the same as we treat lexical items. Results from almost a decade of collostructional analysis have, for the most part, been taken to support the phrasal approach to argument structure, but recently the validity of this claim has been questioned by Müller and Wechsler [14].

There are two issues at hand. The first one concerns the interpretation of collostructional analysis, and whether the basic assumptions made by their proponents are really justified. And secondly, whether results from systematic collostructional analysis does support the phrasal approach to argument structure. I will argue that traditional collostructional analysis is partially wrong in assuming that grammatical constructions can be treated just like lexical items, and that if we apply collostructional analysis to all 'verbal constructions' in a treebank we find strong evidence **against** the phrasal approach to argument structure.

This paper also has the aim of showing how to expand the uses of treebanks for the purpose of linguistic research, beyond that of being repositories for finding examples (Augustinus and Van Eynde [1], Augustinus, Vandeghinste and van Eynde [3], Augistinus, Vandeghinste, Schuuman and van Eynde[2]).

[3] The ARG-ST feature follows the convention that the first argument is the subject, while the rest of the arguments are the other complements of the verb ordered according to the obliqueness hierarchy (Pollard and Sag [15])

2 Collostructional analysis

2.1 Phrasal approach

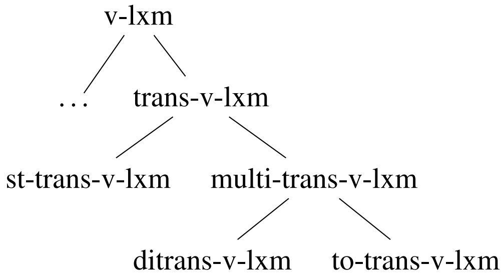
In the initial proposal, collostructions were seen as the grammatical equivalent of collocations (Stefanowitsch and Gries [18]), the same way that a word w_0 can attract with different strengths different collocates $w_1 \dots w_n$, a construction C with a structure [X ...] can attract different words $w_1 \dots w_n$ to its structural position X.

But this view faces consistency problems. If we take the classical *X is waiting to happen*, we could in fact claim that X is attracted by the phrase *is waiting to happen*, just as in normal collocations because both elements are surface signs. However, when we look at argument structure constructions, the picture is a lot less clear. Argument structure constructions are not surface elements. In the classic example of the dative construction [SUBJ VERB DOBJ IOBJ] there is nothing attracting the verbs that occur in the construction, and the attraction is fundamentally asymmetric.

2.2 Lexical approach

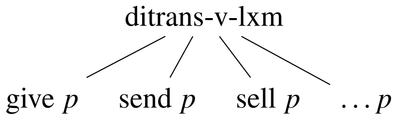
The quantitative effects obtained from carrying out collostructional analysis on argument structure patterns are better expressed as a probability distribution over lexical classes, related by lexical rules or type hierarchies as we find in SBCG and HPSG. A simplified example of the type hierarchy for verbs adapted and slightly modified from Sag et. al. [17] is given in (1):

(1)



We can thus encode probabilities of a verb-lexeme belonging to a particular type.

(2)



Whether probabilities 'percolate' up the tree or are constraint to maximal types is an open issue. In this paper I assume they do percolate, but this is not a crucial assumption.

2.3 Corpus predictions

The result that certain valency patterns were strongly associated with some verbs was taken to mean that these valency patterns had to be phrasal constructions. The rationale behind this was that there could only be attraction if the construction was a real entity. But the existence of attraction between verbs and valency patterns says nothing about the nature of the structures that produce those valency patterns. Despite these claims, there has not been so far a systematic analysis of all (or most) valency patterns in a language. The use of treebanks for collostructional analysis allows us precisely this.

To address the question of phrasal vs lexical constructions we need to test whether the overall collostructional effects are more likely due to a phrasal structuring of the grammar, or a lexical one. I claim that the lexical model makes the following testable predictions: (P1) more general valency patterns (that is, patterns higher in the signature) will be more frequent because they will have a larger portion of the probability space, and (P2) valency patterns that can be related by either a lexical rule of valency augmentation or reduction, or by being sisters in the hierarchy, will be highly correlated in relation to which verbs are likely to be found with them.

Calculating the exact predictions of the the phrasal approach is less straightforward, but it is clear that it does not make these predictions. On the contrary, if phrasal constructions like the transitive construction can attract verbs, the so can the passive, but the passive and transitive constructions can not be linked unless one assumes a high degree of information belongs to the verb (like verb types). More over, phenomena like subject deletion are treated in the phrasal approach as null instantiation constructions, which would in turn have no link to general argument structure constructions. Thus, we should either see no pattern, or null instantiation constructions should be able to attract verbs of their own and exhibit patterns comparable to those found in regular transitive or ditransitive constructions.

3 Methodology

To test (P1) and (P2) I examined all verbs, in all their valency patterns on the section A of the Hamburg dependency treebank[4] (Foth et al. [4]) (a German treebank) which consists of 101,999 manually annotated sentences (around 1.7 million words), checked for consistency. I extracted all verbs and their complements. The valency pattern of a given verb occurrence is then an ordered list of pairs parts-of-speech/syntactic-function of its complements. For all particle verbs I did not

[4] An anonymous reviewer is concerned about the fact that dependency grammar is an inherently lexical theory, and this fact would bias the results or make the analysis outright circular. This is, however, not an issue. The dependency grammar with which the corpus is parsed neither represents HPSG nor Cognitive construction grammar. Even though the corpus is parsed with a lexicalist theory, it does not include any information regarding lexical rules, verb types or actual argument structure, but only surface patterns.

consider the particle to be an argument of the verb, but rather part of it. I took a simplified version of the POS tags because I assume most verbs do not make a strict distinction between all possible kinds of noun phrases and pronouns. This process is the equivalent of performing collostructional analysis on all verbal constructions on the corpus.

4 Results

Figure 1 presents the frequencies of the main 30 valency patterns (out of 177) found in the corpus. As we can see the first 5 valency patterns make up 80% of the data, the first 10 make up 90% of the data, and the 30 displayed make up 98% of the data, leaving the rest a minute portion of the observed patterns. If we observe carefully, the first 5 patterns are also the most general ones: transitives, intransitives in main sentences or selected by an auxiliary. This result is fully consistent with (P1), because these are the valency patterns we would see higher in the grammar signature (intransitives and transitives). More specific and idiosyncratic valency patterns are much more less frequent. A somewhat unexpected result is a large number of (not zipf distributed) different valency patterns. This initial result seems to support (P1), and is easy to capture within the lexical approach. Although not inconsistent with this fact, there is nothing in the arquitecture of phrasal approaches that would predict that some constructions should be more frequent than other[5].

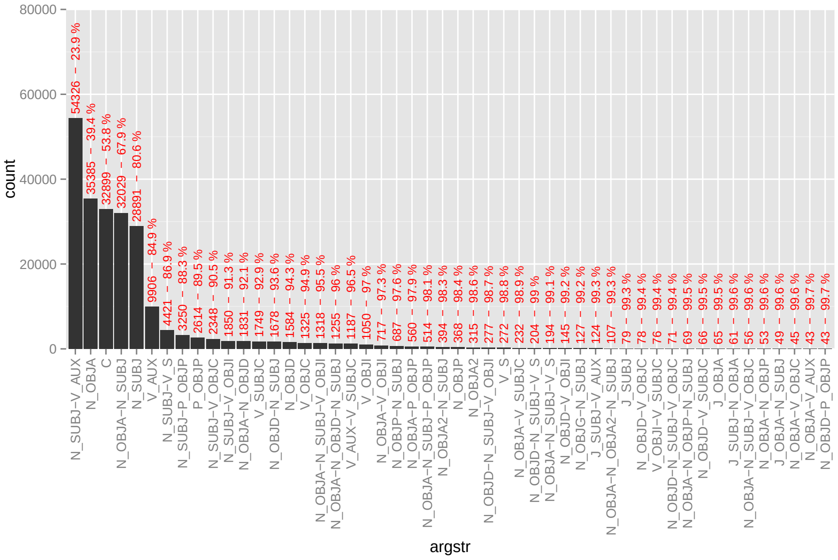


Figure 1: Raw frequencies and cumulative proportions of valency patterns in the corpus. Labels are those of the Hamburg treebank plus: C=no valents, J=adjectives, and A=adverbs

[5] It is of course always possible to assign probabilities to different constructions.

This method allows us to perform a very wide variety of tests and analysis that are not available in normal collostructional analysis. Specially interesting is how the probability space of valency patterns is distributed in the corpus, and their respective entropies (Table 1).

pattern	N	probability	types	entropy
C	32899	0.144	2758	6.498
N_OBJA	35385	0.155	2417	6.477
N_OBJA-N_SUBJ	32029	0.140	1817	5.709
N_OBJD	1584	0.006	281	4.889
V_OBJC	1325	0.005	231	4.634

Table 1: Entropy of different valency patterns

We can also perform the traditional analysis of strongly associated verbs using any significance test we want. In classical collostructional analysis Fisher's exact test or χ^2 are usually employed, but these tests ignore the distribution of the verbs occurring in other constructions. With the present approach we have all the information about those verb's distributions, so different association measurements can be used. Ultimately the question of which association measures are better is an empirical question I will not address, but as an example we can calculate a weighted probability as follows: $WP(v|c) = P(v|c)/H(v)$, where $P(v|c)$ is the probability of a verb in a construction, that is, the number of co-ocurrences divided by the total number of occurrences of the construction, and H(v) is the entropy of the verb, calculated with respect to the verb's dispersion across constructions. As an example we calculate the most attracted collexemes to the dative valency pattern N_OBJA-N_OBJD-N_SUBJ (Table 2). This measure shows actually high correlation with p-values obtained using Fisher's exact test (r = 0.78, p<0.00001).

verb	gloss	v&c	total(v)	P(vlc)	WP(v)	H(v)
versprechen	promise	61	279	0.04860558	0.10285109	2.116035
geben	give	80	3196	0.06374502	0.09719535	1.524752
stellen	set/put	70	1474	0.05577689	0.09136794	1.638097
bieten	ask for	94	1203	0.07490040	0.08889766	1.186878
werfen	throw	44	226	0.03505976	0.06626448	1.890044

Table 2: Most strongly associated verbs to the ditransitive valency pattern.

Finally, to test (P2) we can investigate the correlations between all pairs of the 10 most frequent valency patterns (for reasons of space I have to limit this analysis to only 10 patterns). That is, we build a correlation matrix according to how many times each verb occurs with each valency pattern.

What we see in Figure 2 is that the strongest correlations are between valencies related by direct "deletion", and more precisely, by deletion of the subject. The 3 main correlations (>0.7) are between verbs without dependents and intran-

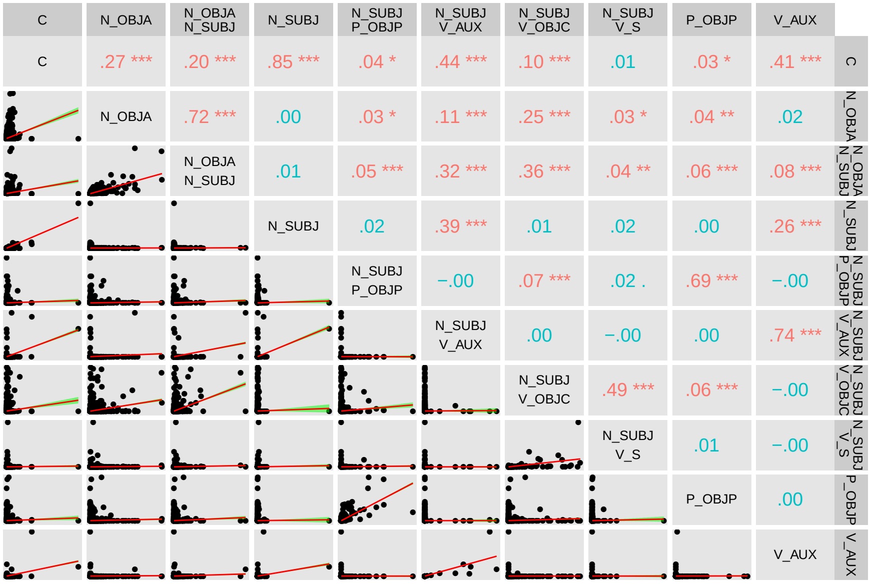


Figure 2: Correlations between the 10 most frequent verb valencies. Red magnitudes indicate statistically significant correlations. Stars indicate the p-values.

sitive sentences (N_SUBJ), verbs with only accusative (N_OBJA) objects and fully transitive sentences (N_SUBJ - N_OBJA), and finally between verbs with only an auxiliary as a complement (V_AUX) and those with a subject and an auxiliary complement (N_SUBJ - V_AUX).

Similarly, we can perform clustering analysis on correlation distance. We can see the result on Figure 3[6]. What this figure shows us is that many correlations (although not all of them) are heavily representative of what we would consider to be verb types with a close or identical argument structure. For example cluster (3) contains basically verbs that select at least one accusative object, and some that also select a dative object. Cluster (15) has verbs that select an accusative object and a genitive object. Cluster (11) verbs with both a prepositional object and a non-finite verb object. Cluster (8) verbs with double accusative objects. And cluster (6) verbs with prepositional objects.

These results make perfect sense from a lexicalist perspective on argument structure, because deletion of certain arguments can be easily handled by a lexical rule. It is, however, not clear how a phrasal approach could explain these observations, since there is nothing that directly links a [X . . .] phrasal construction with a [X . . . Y] construction (at least not in a way that does not make the phrasal approach a just a notational variant of the lexical approach). Null instantiation constructions would also not work for any of the clusters we observe because these would in turn

[6] For the cutting algorithm see Langfelder et al. [12].

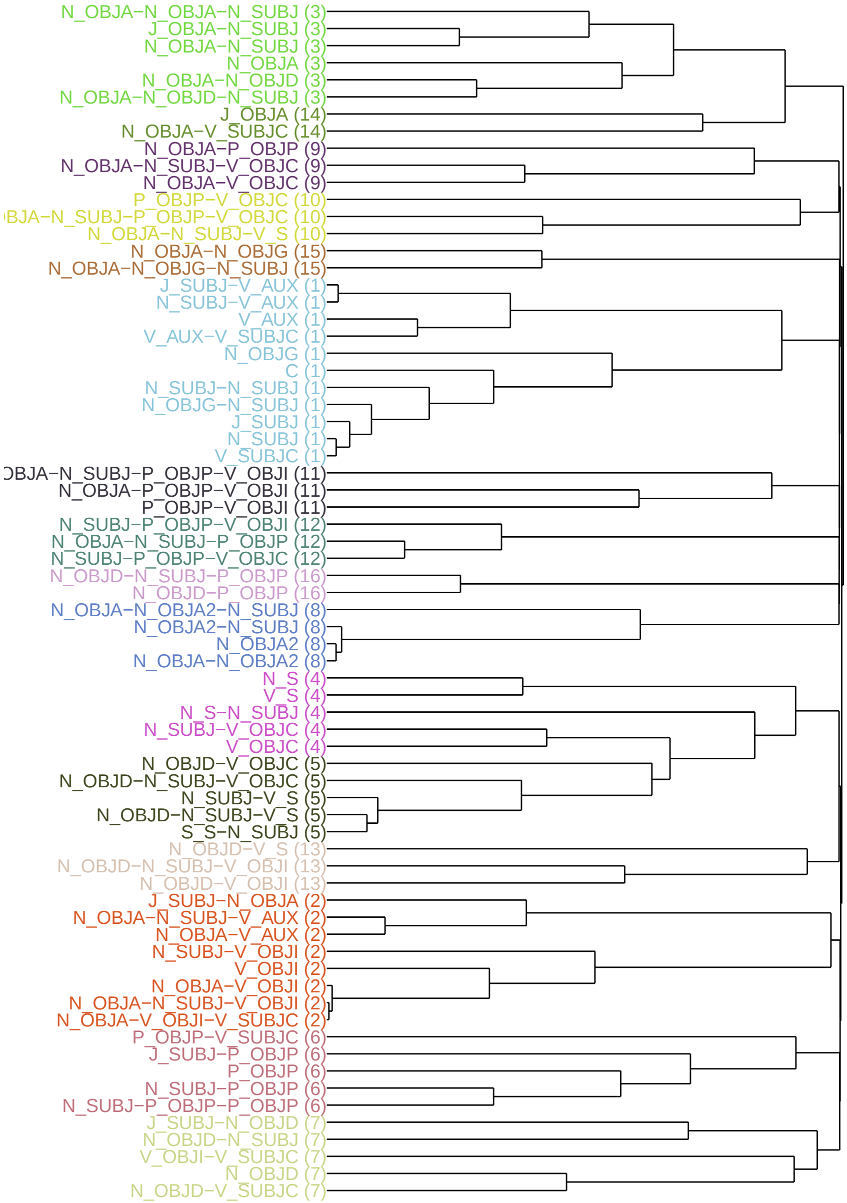


Figure 3: Clustering of the 70 most frequent valency patterns based on correlation scores. Numbers in brackets indicate cluster in order of cluster size. Clusters were extracted with a dynamic tree cutting algorithm.

attract verbs just like a transitive or ditransitive construction[7], which would lead us to expect null subject patterns to be highly correlated with each other, but we do not see this. The only alternative would be to introduce transformations, which seems highly undesirable for a construction grammar.

5 Conclusion

Treebanks are clearly being underused by both theoretical linguists, and corpus oriented linguists. The present paper shows how we can improve corpus linguistic methods like collostructional analysis with the use of treebanks, and how these can be used to explore important theoretical questions.

References

[1] Liesbeth Augustinus and Frank Van Eynde. A treebank-based investigation of ipp-triggering verbs in dutch. *Proceedings of the Eleventh International Workshop on Treebank and Linguistic Theories (TLT11)*, pages 7–12, 2012.

[2] Liesbeth Augustinus, Vincent Vandeghinste, Ineke Schuurman, and Frank Van Eynde. Example-based treebank querying with gretel–now also for spoken dutch. *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, pages 423–428, 2013.

[3] Liesbeth Augustinus, Vincent Vandeghinste, and Frank Van Eynde. Example-based treebank querying. In *LREC*, pages 3161–3167. Citeseer, 2012.

[4] Kilian Foth, Arne Köhn, Niels Beuck, and Wolfgang Menzel. Because size does matter: The hamburg dependency treebank. In *Proceedings of the Language Resources and Evaluation Conference 2014 / European Language Resources Association (ELRA)*. Universität Hamburg, 2014.

[5] Jonathan Ginzburg and Ivan Sag. *Interrogative investigations*. Stanford: CSLI publications, 2000.

[6] Adele E Goldberg. *Constructions: A Construction Grammar Approach to Argument Structure*. Chicago University Press, 1995.

[7] Adele E Goldberg. *Constructions at work: The nature of generalization in language*. Oxford University Press, 2006.

[8] Adele E Goldberg. Fitting a slim dime between the verb template and argument structure construction approaches. *Theoretical Linguistics*, 40(1-2):113–135, 2014.

[7] Unless we assume null instantiation constructions are radically different from regular argument structure constructions. But it is unclear how this would work or how this would be enforced.

[9] Adele E Goldberg and Ray Jackendoff. The english resultative as a family of constructions. *Language*, pages 532–568, 2004.

[10] Stefan Th Gries and Anatol Stefanowitsch. Extending collostructional analysis: A corpus-based perspective onalternations'. *International journal of corpus linguistics*, 9(1):97–129, 2004.

[11] Paul Kay. Unary phrase structure rules and the cognitive linguistics lexical linking theory. *Theoretical Linguistics*, 40(1-2):149–163, 2014.

[12] Peter Langfelder, Bin Zhang, and with contributions from Steve Horvath. *dynamicTreeCut: Methods for detection of clusters in hierarchical clustering dendrograms.*, 2014. R package version 1.62.

[13] Stefan Müller. Phrasal or lexical constructions? *Language*, pages 850–883, 2006.

[14] Stefan Müller and Stephen Wechsler. Lexical approaches to argument structure. *Theoretical Linguistics*, 40(1-2):1–76, 2014.

[15] Carl Pollard and Ivan A Sag. *Head-driven phrase structure grammar*. University of Chicago Press, 1994.

[16] Ivan Sag and Thomas Wasow. Performance-Compatible Competence Grammar. In K. Börjars R. D. Borsley, editor, *Non-Transformational p Syntax: Formal and Explicit Models of Grammar*, pages 359–377. Wiley, 2011.

[17] Ivan Sag. Sign-Based Construction Grammar: An Informal Synopsis. In Ivan A. Sag Hans C. Boas, editor, *Sign-Based Construction Grammar*, pages 69–202. University of Chicago Press, 2012.

[18] Anatol Stefanowitsch and Stefan Th Gries. Collostructions: Investigating the interaction of words and constructions. *International journal of corpus linguistics*, 8(2):209–243, 2003.

On an Apparent Freedom of Czech Word Order. A Case Study

Kateřina Rysová, Jiří Mírovský and Eva Hajičová

Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University in Prague
E-mail: {rysova|mirovsky|hajicova}@ufal.mff.cuni.cz

Abstract

The aim of the present contribution is to document, on the material of the Prague Dependency Treebank (PDT), that the assumed freedom of Czech word order is not really a freedom but that it is guided by certain principles, different from the grammatically given principles determining the word order in some other European languages such as English, German or French. After a short introductory section summarizing the traditional views on Czech word order (Section 1) we briefly characterize our approach to the information structure of the sentence (TFA) and its representation in the annotated corpus of Czech (PDT, Section 2). In Section 3 we present the results of the automatic procedure for dividing the sentence into its topic and focus part and compare these results with human annotators decisions. In Section 4 we put forward and test the hypothesis on the order of elements in the focus part of the sentence, adding some observations that follow from the tests. A summary of our investigations is given in Section 5.

1 Traditional views on Czech word order

Since the pioneering studies of Vilém Mathesius [17] topic–focus articulation is considered to be the primary factor determining the word order in Czech; other factors influencing the Czech word order are then considered to be secondary, be it the grammatical or the rhythmical factors (see [5, p. 609]). According to Uhlířová [23], among the word order principles governed by grammar there is the basic position of a congruent attribute before the governing noun and the modificiation of manner before the verb; grammatical as well as rhythmical aspects govern the position of the so-called clitics, typically the second sentence position, i.e. the position after the first member of the sentence carrying the stress (the so-called Wackernagel position). A special attention is paid to the position of the so-called rhematizers; their position is closely related to their scope.

2 Topic–focus articulation of the sentence and its representation in PDT

2.1 Topic–focus articulation

Our investigation follows the **theoretical account of topic–focus articulation** (TFA in the sequel, see e.g. [21], [12]) within the framework of the Functional Generative Description, according to which the sentence can be divided into what the sentence is **about** (its topic) and what it says **about the topic** (its focus). It is assumed that the **dichotomy of topic and focus** (which is supposed to be very important especially for the specification of the scope of negation) is based on the primary notion of contextual boundness.

2.2 TFA in the Prague Dependency Treebank

The **empirical material** we base our analysis on is the (mostly) manually annotated corpus of Czech, the **Prague Dependency Treebank** (PDT, [2]). TFA is captured there by means of a special attribute of TFA assigned to (almost) each node of the deep structure dependency tree (so-called tectogrammatical level) which may obtain one of the three values: *t* for a non-contrastive contextually bound node, *c* for a contrastive contextually bound node and *f* for a contextually non-bound node.[1] It is assumed that the verb stands on the boundary between topic and focus. The left-to-right dimension of a tectogrammatical tree serves as the basis for the specification of the scale of communicative dynamism: communicative dynamism is understood as the deep word order, with the dynamically lowest element standing in the leftmost position and the most dynamic element (the focus proper of the sentence) as the rightmost element of the dependency tree.

3 Automatic procedure of topic–focus division

3.1 Algorithm for topic–focus division

The algorithm determining the topic–focus boundary is based on the information on the contextual boundness for each node of the tectogrammatical tree and takes into account the status of the main verb (i.e. the root) of the sentence and its immediate dependents.[2] Basically, the algorithm (referred to in the sequel as SH algorithm) includes the following steps ([22], [20]):

[1] There are 206,537 tectogrammatical nodes annotated as contextually bound, out of them 30,312 are contrastive contextually bound. Further, 354,841 nodes are contextually non-bound and for 41,332 nodes, contextual boundness is not annotated (e.g., coordinating nodes, nodes inserted for technical reasons etc.).

[2] Another algorithm (see [14]) for the identification of the boundary between topic and focus in Czech, different from the one discussed below, was based on the position of the verb and the order of verb complementations following the verb compared to the hierarchy of systemic ordering (this hierarchy is discussed below in Section 4).

(a) the main verb (V) and any of its direct dependents belong to focus iff they have the TFA value index *f*;

(b) every item that does not depend directly on V and is subordinated to an element of focus as determined by (a), belongs to focus (where "subordinated to" is defined as the irreflexive transitive closure of "depend on");

(c) iff V and all items directly depending on V are *t* or *c*, then follow the rightmost edge leading from the main verb to the first node(s) on this path that is/are contextually non-bound; this/these node(s) and all the nodes subordinated to it/them belong(s) to focus;

(d) every item not belonging to focus according to (a) – (c) belongs to topic.

3.2 The original implementation

Zikánová et al. [25] implemented and applied the SH algorithm to a part of the PDT data (about 11 thousand sentences). The results of their implementation indicate that a clear division of the sentence into topic and focus according to the hypothesized rules has been achieved in 94.28% of sentences to which the procedure has been applied; 4.41% of sentences contained the type of focus referring to a node (or nodes) that belong(s) to the communicatively most dynamic part of the sentence though they depend on a contextually bound node. The real problem of the algorithm then rests with the case of ambiguous partition (1.14%) and cases where no focus was recognized (0.11%).

To validate the algorithm, the same PDT documents were analyzed manually, most of them in three parallel annotations (about 10 thousand sentences), and about 600 sentences in six parallel annotations [26]. The annotators (mostly high school students, having some basic idea of the dichotomy of topic and focus as "the aboutness relation" but not being familiar with the theoretical framework of TFA) worked with the raw texts and were instructed to mark – according to their understanding of the given sentence – every single word in the sentence as belonging either to topic or to focus, and, in addition, to mark which continuous part of the sentence they understand as topic and which continuous part as focus. One of the important subtasks of this project was to follow annotators' agreement/disagreement. The disagreement in the assignments of the two parts of the sentence as a whole was rather high and indicates that the intuitions concerning the division of the sentence into its topic and focus parts may dramatically differ. It is interesting to note that the annotators' agreement in the assignments of individual words in the sentences to topic or to focus was much higher (75%) than the agreement in the assignment of the topic-focus boundary (36%) in both the three and six parallel analyses. Most of the cases in which the annotators disagree concerned the position of the verb in the topic or in the focus. It should be also taken into consideration that while we get only a single, unambiguous result from the automatic procedure, more ways of interpretation could be correct.

3.3 New experiment setting and implementation of the algorithm

In the present study, we evaluate the SH algorithm for the division of the sentence into its topic and focus part presented above in Section 3.1 in a slightly different way. First of all, we tried to avoid some of the shortcomings of the implementation of the algorithm reported in [25], most importantly the inability to properly treat coordination structures. Another difference was that we used data annotated by a linguistic expert as the gold data rather than those used in [23] as a result of voting based on the agreement/disagreement of annotators. Our assumption was that both these changes in the experiment setting would bring an improvement in the results and would better reflect the adequacy of the algorithm for transforming values of contextual boundness into the division of the sentence into the topic and the focus.[3]

Our gold data consisted of 319 sentences from twelve PDT documents annotated by a single linguistic expert familiar with the topic–focus articulation theory. The annotation proceeded directly on the tectogrammatical trees, in an annotation environment adjusted for the task. Without taking into account (already annotated but now hidden) values of contextual boundness, the annotator marked each tectogrammatical node as belonging either to the topic or to the focus.

On these gold data, we evaluated our implementation[4] of the SH algorithm, as well as a baseline algorithm similar to the baseline used in [25].[5] However, in our case both the implementation of the SH algorithm and the implementation of the baseline algorithm took into account coordination structures and the rules were applied to each coordinated member separately. Similarly to [25], we did not count all surface sentence tokens in the evaluation, but rather – in our setting – only nodes relevant for topic–focus articulation assignment.[6]

Table 1 shows a comparison of our implementation of the baseline algorithm and our implementation of the SH algorithm. Our implementation of the SH algorithm significantly outperforms the baseline in most of the measured phenomena.

At the same time, both our implementations of the baseline algorithm and of the SH algorithm outperform most of the results presented in [25]; their F1-measure in focus was 0.88 for the baseline and 0.83 for their implementation of the SH algorithm (they did not report their results in topic). The results are strictly speaking not directly comparable, as the implementations were evaluated on different data and in slightly different experiment settings. However, if we exclude the proper treatment of coordinations in our implementation (thus getting closer to the implementation used in [25]), our results for F1-measure in focus drop to 0.8 for the

[3] By using the expertly annotated data we have lost the connection with the language intuition of the non-expert annotators. In this sense, we evaluate the algorithm rather than the agreement between the TFA theory and the intuition of non-expert speakers; that is exactly what we wanted to do.

[4] We use a slightly modified implementation of the SH algorithm programmed in 2007 by Jiří Havelka, which – to our best knowledge – has never been published.

[5] The baseline is defined as follows: in the linear (surface) form of the sentence, each word before the autosemantic part of the predicate verb belongs to topic, the rest of the sentence belongs to focus.

[6] I.e. we only evaluated tectogrammatical nodes that had a value of contextual boundness filled in. It means that for example coordination nodes have been left out of the evaluation.

Measure	Baseline	SH Algorithm
recall in topic	0.69	0.94
precision in topic	0.89	0.85
F1-measure in topic	0.78	0.89
recall in focus	0.96	0.93
precision in focus	0.88	0.97
F1-measure in focus	0.92	0.95
overall accuracy on tectogrammatical nodes	0.88	0.93
overall accuracy on whole sentences	0.31	0.75

Table 1: Evaluation of our implementation of the baseline algorithm and of our implementation of the SH algorithm on our gold data.

baseline algorithm and to 0.88 for the SH algorithm. It means that the proper way of processing coordinations plays an important factor in achieving better results. Apart from this, we attribute the improvement also to the different kind (we believe "better quality") of the gold data on which we measure the results. We find also very important that our implementation of the SH algorithm outperforms the baseline, while in the experiments reported in [25], the results were the opposite.

4 Order of words in the focus part of the sentence

4.1 General remarks

As stated in Section 1, the topic–focus articulation is traditionally understood as the primary factor of Czech word order. However, with a more detailed look at the word order in relation to the topic–focus bi-partition, a substantial difference has come out. The ordering of the elements in the topic part is basically motivated by the degree of activation of the corresponding elements of the stock of knowledge assumed by the speaker to be shared by him and the hearer,[7] while in the focus part there is a close relationship of the ordering of the sentence elements to the syntactico-semantic types of the verb modifications.

The postulation of a certain hierarchy of cognitive entities that is reflected in the ability of the hearer to identify the referents of the expressions used in the sentence and eventually reflected in the order of sentence elements dates back to the eighties of the last century. Based on the notion of "assumed familiarity", a familiarity scale is defined by Prince [18, p. 245]. In a similar vein, Gundel et al. [11] accepts Givon's [6] scale in the syntactic coding of topic accessibility and maintains that the assumed cognitive (memory and attention) status of an intended referent/interpretation is connected with the appropriate use of a different form or

[7] For the notion of this hierarchy see e.g. [15], [13].

forms (articles, demonstrative pronouns). Ariel [1] concentrates on the system of accessing NP antecedents and follows Givon's view that the Accessibility Marking Scale cannot be taken as universal because it does not cover the full range of referring expressions in all languages. Lambrecht [16] bases his analysis of referent accessibility on Chafe's ([3], [4]) idea of three activation states and relates them to their formal correlates in the structure of sentences. A slightly different but yet related is the model of the local attentional states of speakers and hearers as proposed by Grosz and Sidner ([9], [10]), which is the basis of the centering theory ([7], [8], [24]). Discourse entities are ranked according to language-specific ranking principles; the ranking of centers is defined in terms of syntactic relations as specified for the surface shape of the sentence (subject, object etc.) and as such are very language-dependent.

Considerations similar to these have led us to study the word order separately for the topic and for the focus part of the sentence. Apart from a minor influence of grammatical principles mentioned in Section 1, we consider the order of sentence elements in the **topic** part to be basically influenced by the previous co-text and the situational contexts as well as by the intended discourse strategy of the speaker, his interests and also by his intention to put certain elements in contrast to the previous co-text or situation. For the order of elements in the focus part of the sentence, a hypothesis of the so-called systemic ordering was formulated; it is discussed in more detail below.

4.2 The hypothesis of systemic ordering

The original empirical study of Czech texts has led to the assumption [21, p. 69] that the ordering of the elements in the **focus** part of the sentence is primarily given by the type of the complementation of the verb. This hypothesis was empirically tested pairwise (i.e., successively for two of the complementation types) and it was also supported by several psycholinguistic experiments [21, p. 72ff]. It was assumed that systemic ordering is a universal phenomenon and that at least in most European languages the order of the principle verb complementations (such as Actor – Addressee – Patient) is the same. The following detailed ordering has been established for Czech:

Actor – Temporal (when – since when – till when – how long) – Location (where) – Manner – Extent – Measure – Means – Addressee – From where – Patient – To where – Effect – Condition – Aim – Cause

4.3 Testing of the hypothesis

It was clear from the very beginning that the hypothesis of systemic ordering is very **strong** and that further investigation based on a much broader material is needed, which may lead to a more precise specification or modification(s). The material of the **Prague Dependency Treebank** opened the possibility to validate the hypothesis. A rather complex analysis is presented by K. Rysová [19], who arrives (among other important findings) at the following observations:

(i) There is a tendency of a contextually non-bound element expressed by a clause to follow the non-sentential element (which is apparently connected with the 'weight' of the element).

(ii) There is an influence of the form of the complementation: e.g., the assumed order Manner – Patient is more frequent if the complementation of Manner is expressed by an adverb and the complementation of Patient by a nominal group.

(iii) As for the position of the Actor on the scale, a substantial number of counterexamples of the original hypothesis concern cases for which the outer form of the Actor plays an important role: in sentences with the verb *být* [*to be*] in structures of the type *je nutné* (PAT) *přiznat* (ACT) [*it is necessary* (PAT) *to acknowledge* (ACT)], where Actor is expressed by infinitive, Patient precedes Actor, while the hypothesized order Actor – Patient is attested to if both complementations are expressed by nominal groups.

(iv) There is also a possibility that the order might be influenced by the difference in the optional/obligatory character of the given complementations: there is a tendency that obligatory complementations follow the optional ones though this tendency is not a very influential word order factor.

Rysová's analysis confirms that there is a considerable tendency that in such pairs one ordering prevails over the other, which was the starting point of the postulation of the systemic ordering hypothesis. However, with some pairs, such as Patient and Means, there was a balance between the frequency of the two possible orders, which may indicate that for some particular complementations more than a single complementation occupy one position on the scale.

4.4 Further supporting factors

4.4.1 Position of the verb

The data on the annotators' agreement/disagreement on the topic/focus boundary presented above in Section 3.2 have invited a more detailed study of the decisions of the annotators, especially from the point of view of a possible influence of the verb position.

It has been confirmed that in the Czech surface word order the verb (be it contextually bound or non-bound) can be shifted into the **second position** even if followed by contextually bound elements: both Examples (1) and (2) can serve as a reply to *What did Dan do yesterday?* or *Where did Dan go yesterday?*, i.e the verb *jel* [*went*] may be contexually bound or non-bound.

(1) *Dan včera jel z Prahy do Brna.*
[Lit. *Dan yesterday went from Prague to Brno.*]

(2) *Dan jel včera z Prahy do Brna.*
[Lit. *Dan went yesterday from Prague to Brno.*]

In the sample of the PDT studied for this purpose,[8] there are 6,458 sentences (15% cases) with the verb in the second position as compared with 37,208 sentences with the verb on other than the second position. The order of the contextually non-bound complementations of the verb was studied pairwise, with the following results:[9]

(a) The ordering was in accordance with the hypothesized systemic ordering in cases of LOC–PAT, PAT–AIM, TWHEN–ADDR, LOC–EXT, PAT–EFF, TWHEN–PAT, ADDR–PAT, THO–PAT, THL–PAT and TWHEN–LOC; this result holds for the sentences regardless of the verb position in them, see Example (3).

(3) *Termín pseudohumanisté má ostatně v našem politickém životě*.LOC *dlouhou tradici*.PAT
[Lit. *The term pseudohumanists has, after all, in our political life*.LOC *a long tradition*.PAT]

(b) The hypothesized systemic ordering was confirmed also with the pairs EXT–PAT, TWHEN–EXT and MANN–EFF. However, in sentences with the verb on other than the 2nd position, the supposed ordering predominated very strongly, except for the pair EXT–PAT which was present especially with the verb on the 2nd position. For pairs MANN–PAT (see Example (4)) and PAT–DIR3, the systemic ordering was confirmed in sentences with the verb on other than the 2nd position. In cases with the verb on the 2nd position, none of the orderings was prevailing.

(4) *Kromě toho do tří let postaví ve městě na své náklady*.MANN *travnaté fotbalové hřiště*.PAT
[Lit. *In addition, within three years, (they) will build in the city at their expense*.MANN *a grass football field*.PAT]

(c) There was a considerable difference between the cases with the verb on the 2nd position and cases with the verb in other than the 2nd position in the pair PAT–DIR1. In sentences with the verb on other than the 2nd position, the hypothesized ordering DIR1–PAT was present more frequently (see Example (5)), while in sentences with the verb on the 2nd position, the ordering PAT–DIR1 strongly predominated (see Example (6)).

[8] The sentences studied included the verb in the second position in the surface shape of the sentence when the verb was followed by a dependent that, in the corresponding tectogrammatical representation, had the *c* or *t* value of the TFA attribute. Only sentences with indicative mood were taken into account. For testing our hypotheses, 9/10 of the PDT data have been used.

[9] In order to exclude the factor of "weight", we have taken into account only clauses in which the predicate had any number of dependents but the relevant dependents had maximally 3 subordinated nodes.

(5) *Ve starých filmech s Oldřichem Novým hlavy majetných rodin snímaly ze stěn*.DIR1 *obrazy*.PAT*, aby ze schránek za nimi vylovily notářské listiny či rodinné šperky.*
[Lit. *In old movies with Oldřich Nový, the heads of the wealthy families removed from the walls*.DIR1 *paintings*.PAT *in order to find some notarial deeds or the family jewels in the boxes behind them.*]

(6) *Stuttgartská automobilka Porsche přesune letos výrobu*.PAT *ze svého závodu*.DIR1 *v Salcburku do Českého Krumlova.*
[Lit. *The Stuttgart car maker Porsche will move this year the production*.PAT *from its factory*.DIR1 *in Salzburg to Český Krumlov.*]

This analysis has pointed out that in most cases the 2nd position of the verb does not influence the order of the contextually non-bound complementations of the verb. However, the results of this follow-up analysis have also confirmed that the position of ACT in the systemic ordering hierarchy (first on the scale, before all other types of complementations) has to be revised, or at least the contextual conditions for its appearance in other positions in the focus part of the sentence have to be studied in more detail. The pairs that documented this necessity are PAT–ACT, TWHEN–ACT, EXT–ACT, MANN–ACT and THO–ACT. Also the ordering in pair MANN–LOC has to be further analyzed.

4.4.2 Word order in separate clauses

In the original formulation of the systemic ordering as well as in its testing on the PDT no difference was made between the ordering in the **main clause** and in the **dependent clauses**.

We have therefore performed a broader research on testing the hypothesis of systemic ordering with respect to this factor and studied the word order of contextually non-bound verb complementations in the main clauses and in the dependent clauses separately.

This analysis has pointed out that in most cases the sentence character (main clause vs. dependent clause) does not influence the order of the contextually non-bound complementations of the verb within the respective clauses. The main results of the analysis are as follows:

(a) The ordering was in accordance with the hypothesized systemic ordering in case of TWHEN–ADDR, ACT–ADDR, PAT–AIM, MANN–EFF and DIR1–PAT; this result holds both for main clauses and for dependent clauses.

(b) The hypothesized systemic ordering was confirmed also in pairs PAT–EFF, EXT–PAT and TWHEN–EXT in both types of clauses. Pairs PAT–EFF and EXT–PAT occurred in dependent clauses significantly more often, while TWHEN–EXT occurred rather in main clauses. The supposed ordering PAT–DIR3 and LOC–EXT was confirmed only in main clauses. In dependent clauses, the ordering in these pairs was nearly balanced.

As in Section 4.4.1, the analysis has also indicated that the position of ACT in the systemic ordering hierarchy has to be studied in more detail, considering the frequently occurring pairs PAT–ACT and MANN–ACT. Also the ordering in pairs PAT–MEANS and MANN–LOC has to be further analyzed.

We have also followed the appurtenance of the dependent clause to the topic or to the focus part of the sentence, i.e. whether the main predicate of the given dependent clause was contextually bound or non-bound. Unfortunately, the number of contextually bound dependent clauses was very small when compared to the number of dependent clauses that are part of focus; out of 2,861 dependent clauses (fulfilling the above restrictions) there were only 141 dependent clauses the predicate of which had the value of the TFA attribute *t* or *c* (i.e. contextually bound). This low occurrence of dependent clauses in the topic part of the sentence does not allow to draw any conclusions on the order of the verb complementations in them, except for the statement that our data indicate that most dependent clauses occur in the focus part of the sentence.

5 Summary

Testing the relation between TFA and Czech word order on the annotated corpus of Czech, the following assumptions have been confirmed and the following tendencies have been observed:

(a) Czech word order is not ultimately free but it is guided by **communicatively** given principles.

(b) The order of elements in the focus part of the sentence reflects in principle the hypothesis of **systemic ordering**.

(c) There are several factors influencing systemic ordering following from the **syntactic** structure of the sentence and the **lexical** properties of sentence elements.

(d) In most cases, neither the sentence character (the **main** clause **vs. dependent** clause) nor the position of the governing verb in the sentence (the position on the **2nd place vs. other positions**) influence the order of the contextually non-bound complementations of the verb within the sentence.

Our **implementation of the SH algorithm** for the division of the sentence into topic and focus based on the values of contextual boundness outperforms previously published results and, also very importantly, outperforms the baseline.

Acknowledgements

We gratefully acknowledge support from the Ministry of Education of the Czech Republic (project LM2010013 – LINDAT/CLARIN) and the Czech Science Foundation (GA CR, project P406/12/0658). This work has been using language resources developed, stored and distributed by the LINDAT/CLARIN project.

References

[1] Ariel M. (1990). *Accessing Noun-Phrase Antecedeents.* London: Routledge.

[2] Bejček E., E. Hajičová, J. Hajič, P. Jínová, V. Kettnerová, V. Kolářová, M. Mikulová, J. Mírovský, A. Nedoluzhko, J. Panevová, L. Poláková, M. Ševčíková, J. Štěpánek and Š. Zikánová (2013). *Prague Dependency Treebank 3.0.* Data/software, Univerzita Karlova v Praze, MFF, ÚFAL, Prague.

[3] Chafe W. (1979). The flow of thought and the flow of language. In: Givon (ed.), *Syntax and Semantics: Discourse and Syntax, Vol. 12.* New York: Academic Press, 159–182.

[4] Chafe, W. L. (1987). Cognitive constraints on information flow. In: R. S. Tomlin (ed.), *Coherence and grounding in discourse.* Amsterdam: Benjamins, 21–52.

[5] Daneš F., H. Běličová, M. Čejka, E. Dvořák, M. Grepl, K. Hausenblas, Z. Hlavsa, J. Hoffmannová, J. Hrbáček, J. Chloupek, P. Karlík, E. Macháčková, O. Müllerová, B. Palek, J. Nekvapil, J. Novotný, P. Piťha, H. Prouzová, M. Rulfová, B. Rulíková, O. Šoltys, L. Uhlířová and S. Žaža (1987). *Mluvnice češtiny. 3. Skladba* [*Grammar of Czech. 3. Syntax*]. Prague: Academia.

[6] Givon T. (1983). Topic continuity in discourse: An Introduction. In: Givon (ed.), *Topic Continuity in Discourse: A Quantitative Cross-Language Study.* Amsterodam: John Benjamins, 1–41.

[7] Grosz B., A. K. Joshi and S. Weinstein (1983). Providing a unified account of definite noun phrases in discourse. In: *Proceedings of the Annual Meeting of the Association for Computational linguistics 21,* 44–50.

[8] Grosz B., A. K. Joshi and S. Weinstein (1995). Centering: A Framework for modeling the local coherence of discourse. *Computational Linguistics 21,* 203–225.

[9] Grosz B. J. and C. L. Sidner (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics 12,* 175–204.

[10] Grosz B. J. and C. L. Sidner (1998). Lost intuitions and forgotten intentions. In: M. Walker, A. Joshi and E. Prince (eds.), *Centering in Discourse.* Oxford University Press, 39–51.

[11] Gundel J. K., N. Hedberg and R. Zacharski (1993). Cognitive status and the form of referring expressions in discourse. *Language 69,* 274–307.

[12] Hajičová E., B. Partee and P. Sgall (1998). *Topic–Focus Articulation, Tripartite Structures and Semantic Content.* Dordrecht: Kluwer.

[13] Hajičová E. (1993). *Issues of Sentence Structure and Discourse Patterns.* Prague: Charles University Press.

[14] Hajičová E., H. Skoumalová and P. Sgall (1995). An Automatic Procedure for Topic–Focus Identification. *Computational Linguistics, 21(1)*, 81–94.

[15] Hajičová E. and J. Vrbová (1982). On the role of the hierarchy of activation in the process of natural language understanding. In: Horecký J. (ed.), *Coling 82 – Proceedings of the Ninth International Congress of Computational Linguistics,* Prague – Amsterdam, 107–113.

[16] Lambrecht K. (1994). *Information Structure and Sentence Form. Topic, Focus and the Mental Representations of Discourse Referents.* Cambdridge: Cambridge University Press.

[17] MathesIus V. (1941). Základní funkce pořádku slov v češtině. [Basic functions of word order in Czech.] *Slovo a slovesnost 7,* 169–180.

[18] Prince E. (1981). Toward a taxonomy of given/new information. In: Cole P. (ed.), *Radical Pragmatics.*. New York, Academic Press, 223–254.

[19] Rysová K. (2014). *O slovosledu z komunikačního pohledu.* [*On Word Order from the Communicative Point of View.*] Prague: Institute of Formal and Applied Linguistics.

[20] Stall P. (1981). Towards a Definition of Focus and Topic. In: *Prague Studies in Mathematical Linguistics 7.* Prague: Academia, 173–197.

[21] Sgall P., E. Hajičová and E. Buráňová (1980). *Aktuální členění věty v češtině.* [*Topic–Focus Articulation in Czech.*] Prague: Academia.

[22] Sgall P. and E. Hajičová (1977). Focus on focus. In: *The Prague Bulletin of Mathematical Linguistics, No. 28,* 5–54, and *No. 29* (1978), 23–41. Prague: Univerzita Karlova v Praze.

[23] Uhlířová L. (1987). *Knížka o slovosledu.* [*A book on word order.*] Prague: Academia.

[24] Walker M. A., A. Joshi and E. Prince (1998). Centering in naturally occurring discourse: An overview. In: Walker M. A., A. Joshi and E. Prince (eds.), *Centering theory in discourse.* Oxford: Clarendon, 1–28.

[25] Zikánová Š. and M. Týnovský (2009). Identification of Topic and Focus in Czech: Comparative Evaluation on Prague Dependency Treebank. In: Zybatow, Gerhild et al. (eds.), *Studies in Formal Slavic Phonology, Morphology, Syntax, Semantics and Information Structure.* Formal Description of Slavic Languages 7, Frankfurt am Main: Peter Lang, 343–353.

[26] Zikánová Š., M. Týnovský and J. Havelka (2007). Identification of Topic and Focus in Czech: Evaluation of Manual Parallel Annotations. In: *The Prague Bulletin of Mathematical Linguistics, No. 87.* Prague: Univerzita Karlova v Praze, 61–70.

Treebank Data and Query Tools for Rare Syntactic Constructions

Erhard Hinrichs, Daniël de Kok and Çağrı Çöltekin

Department of Linguistics
University of Tübingen
E-mail: {erhard.hinrichs|daniel.de-kok|cagri.coeltekin}
@uni-tuebingen.de

Abstract

This paper reports on the use of the treebank query tool TüNDRA, which is able to process large treebanks of the size needed for rare syntactic constructions such as the Zwischenstellung of finite auxiliaries in German subordinate clauses. The TüPP-D/Z, an automatically annotated treebank with 11.5M sentences, contains a total of 92 examples of this construction, which need to be hand-filtered from corpus queries that produce a significant number of false positives. The corpus findings about the Zwischenstellung shed new light on the usage of this construction in contemporary German that contradict previous claims put forth in the linguistics literature.

1 Introduction

Treebanks serve a variety of purposes in computational linguistics – as training materials for statistical parsers and other automatic language processing tools – and in theoretical linguistics alike. For linguistic research, they provide authentic language materials for linguistic structure in general and (morpho-)syntax in particular. Authentic language materials present an important data type that can supplement grammaticality judgements of native speakers and that can provide valuable information about the actual usage patterns of linguistic constructions across speakers of a language.

The frequency of a particular grammatical phenomena under consideration determines the amount of corpus/treebank data that are necessary for a meaningful empirical investigation. If the phenomenon is relatively rare, then the amount of annotated data may have to be considerable and may go beyond what can reasonably be offered by treebanks such as the Penn Treebank (4.5 million English words;[10]) and the TüBa-D/Z (95.595 sentences with 1.787.801 German word tokens for Release 10.0 (08/2015);[17]) which were produced entirely by manual annotation. Rather, larger treebanks that were constructed semi-automatically or without any manual post-editing such as the TüPP-D/Z [15] may need be to be consulted. The critical mass of data for a given grammatical phenomenon has repercussions not only for the method of annotation, but also for search interfaces that can be used to query treebanks. Most query tools currently only support treebanks up to a certain size, due to performance restrictions of the underlying search algorithms. In addition, since the treebank data are generated entirely by automatic means, the resulting data are noisy. This noisiness has to be taken into account when searching the treebank and when interpreting the results.

The purpose of the present paper is to investigate the so-called *Zwischenstellung* of finite auxiliaries in German as a case study of a low-frequency syntactic construction of German that requires large amounts of data and hence a highly performant query tool. The case study highlights: (i) the importance of verifying the claims that have been made in the linguistics literature about this construction by treebank data, and (ii) the processing requirements imposed on a treebank query tool that can accommodate the required amount of data. More specifically, the TüPP-D/Z will be used as the underlying treebank (see Section 3 below), whose annotations were produced by a finite-state chunk parser, and the TüNDRA [12] web application (see Section 4 below) will be used as the query tool of choice.

2 The Data: Placement of Finite Auxiliaries in German Subordinate Clauses

In subordinate clauses of German, finite verbs usually appear in clause-final position, as in (1a). However, when forms of the auxiliary verb *haben* govern a modal verb such *können* or *müssen* in (1b), then the auxiliary appears leftmost in the verbal complex in the so-called *Oberfeld* – in the terminology of Bech [2] – and the modal verbs are realized as so-called *Ersatzinfinitive* ('substitute infinitives'). The ungrammaticality of (1c) and (1d) show that Oberfeld placement and the use of the Ersatzinfinitiv (instead of the ordinary past participles) are obligatory.

(1) a. dass Eike gesungen hat.
that Eike sung has.
'that Eike has sung.'

b. dass Eike hat singen { können / müssen }.
that Eike has sing { can / must }.
'that Eike was able to / had to sing.'

c. * dass Eike singen { können / müssen } hat.
that Eike sing { can / must } has.

d. * dass Eike kommen { gekonnt / gemusst } hat.
that Eike come { can / must } has.

Examples (2) shows that Oberfeld placement is triggered not only by modal verbs, but also by the verb *lassen* ('let'). However, for *lassen*, clause-final placement and Oberfeld placement of the finite auxiliary are both acceptable, as are the use of the past participle and the Ersatzinfinitiv for *lassen* in the case of clause-final placement of the auxiliary.

(2) dass sie ihn { arbeiten gelassen hat / arbeiten lassen hat / hat arbeiten
that she him { work let has / work let has / has work
lassen }.
let }.
'that she let him work.'

Oberfeld placement of finite auxiliaries is not restricted forms of *haben*, but also occurs with forms of *werden* 'will' in the future tense, as the examples in (3) show.

(3) a. dass sie { arbeiten können wird / wird arbeiten können }.
that she { work be able to will / will work be able to }.
'that she will be able to work.'

b. dass Eike hat singen { können / müssen }.
that Eike has sing { can / must }.
'that Eike was able to / had to sing.'

In examples (1) – (3), the finite auxiliary appears either in initial or final position in the verb cluster. Den Besten and Edmondson [4] have pointed out that there are also cases, where finite auxiliaries appear in the middle of the verbal complex in a so-called *Zwischenstellung* ('intermediate position'), i.e. to the right of the main verb and to the left of the non-finite auxiliary in examples such as (4).

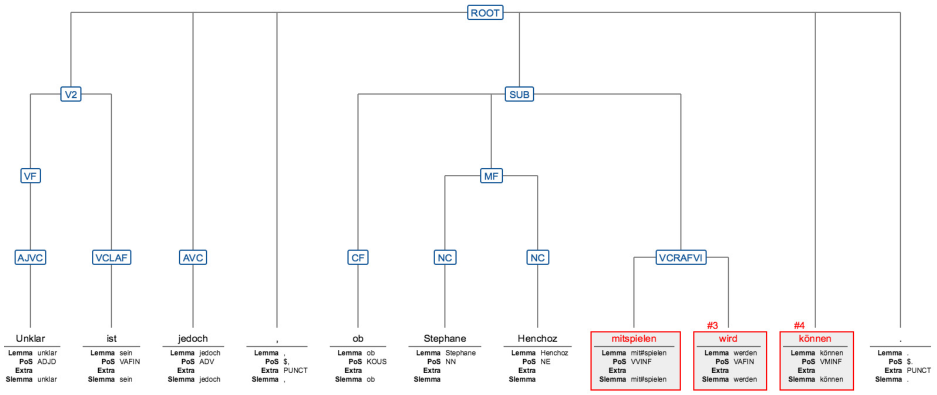
(4) a. dass er arbeiten hat können.
that he work has been able to
'that he has been able to work.'

b. dass er arbeiten wird können.
that he work will be able to
'that he will be able to work.'

c. dass er gewählt hätte werden können.
that he elected had [Passive werden] can.
'that he would have been able to be elected.'

d. dass er abgewählt wird werden können.
that he voted out will [Passive werden] can
'that he will possibly be voted out of office.'

For reasons of space, the data survey of Oberfeld placement of finite auxiliaries is far from complete. It covers only those triggering verbs that are directly relevant for the discussion of the Zwischenstellung in Section 5 below. A more comprehensive account of the Oberfeld is presented, inter alia, in [1], [2], and [5]. The grammaticality judgments on Oberfeld placement reported in this paper or taken


ROOT
V2
SUB
VF
MF
AJVC
VCLAF
AVC
CF
NC
NC
VCRAFVI
#3
#4
Unklar
Lemma unklar PoS ADJD Extra Slemma unklar
ist
Lemma sein PoS VAFIN Extra Slemma sein
jedoch
Lemma jedoch PoS ADV Extra Slemma jedoch
,
Lemma , PoS $, Extra PUNCT Slemma ,
ob
Lemma ob PoS KOUS Extra Slemma ob
Stephane
Lemma Stephane PoS NN Extra Slemma
Henchoz
Lemma Henchoz PoS NE Extra Slemma
mitspielen
Lemma mit#spielen PoS VVINF Extra Slemma mit#spielen
wird
Lemma werden PoS VAFIN Extra Slemma werden
können
Lemma können PoS VMINF Extra Slemma können
.
Lemma . PoS $. Extra PUNCT Slemma .

Unklar ist jedoch, ob Stephane Henchoz mitspielen wird können.
unclear is however whether Stephane Henchoz play with will can
'It is unclear, however, whether Stephane Henchoz will be able to play.'

Figure 1: TüPP-D/Z sentence with Zwischenstellung of *wird*

from [5]; however, see [7] for a dissenting view on the acceptability of Oberfeld formation with *werden* and *lassen*, as in (3).

3 The Corpus

The TüPP-D/Z (Tübingen Partially Parsed Corpus of Written German[1] treebank uses as its data source the Scientific Edition of the taz German daily newspaper[2], which includes articles from September 2, 1986 up to May 7, 1999. The corpus consists of 11,512,293 sentences with a total of 204,425,497 tokens. The texts are processed automatically, starting from paragraph, sentence, word form, and token segmentation. All sentences have been automatically annotated with clause structure, topological fields, and chunks, as well as parts of speech and morphological ambiguity classes. Figure 1 shows a sentence from the TüPP-D/Z with Zwischenstellung of the finite auxiliary *wird* ('will') in the verbal complex of a subordinate clause, which is headed by the clause label SUB. The subordinate clause and the main clause form the root clause (ROOT) and are annotated by topological field labels. Main clauses in German place the finite verb in second position; hence the clause label V2. Topological field annotation for main clauses include the Vorfeld (VF) and the left bracket (VCL) with the finite verb. Since the finite verb *ist* in Figure 1 is a finite auxiliary (FA), the left bracket is identified as VCLAF. The topological field structure of a German subordinate clause includes the complementizer field (CF) as the left bracket and the verbal complex as the right bracket (VCR). In Figure 1, the verbal complex is realized as a finite auxil-

[1] www.sfs.uni-tuebingen.de/ascl/ressourcen/corpora/tuepp-dz.html
[2] www.taz.de

iary *wird* in Zwischenstellung and a non-finite main verb (VI). Accordingly, the verbal complex label is further specified as VCRAFVI. As will be described in Section 5, use to this topological field label in search queries for Zwischenstellung examples for the TüPP-D/Z.[3]

4 Querying Large Treebanks

TüNDRA [12] is a web applications that allows linguists to search and visualize treebanks. It uses the TIGERSearch [8] query language, with support for existential negation. Moreover, it supports both constituency trees and dependency graphs. Recently, the back-end of TüNDRA was rewritten to support large treebanks in the order of hundreds of millions words [3], such as used in this study.

4.1 Query Processing

Treebank search tools use a variety of different query engines and storage methods. Storage methods run the gamut from formats specific to treebank applications to generic graph databases. Specific storage formats provide more opportunity for optimization for the task at hand, whereas using a generic solution allows a treebank tool to leverage existing well-tested storage systems that typically support widely-used standards. We will give an example of both opposites of this gamut.

- INESS-Search [13] uses an on-disk format that is specifically developed for (directed graph) treebanks. It uses inverted indices for the features that are represented in the treebank (such as *word*, *cat*, *parent-edges*, and *child-edges*). The lexicon-part of the indices is stored as a suffix array [9], allowing for quick lookup of sentences and nodes using regular expressions. INESS-Search uses an extended version of the TIGERSearch query language. Queries are parsed to an internal representation that is similar to the logical form of the query. The inverted indices and relation/predicate signatures are used to restrict the set of candidate nodes. As Meurer [13] points out, the use of task-specific storage eliminates overhead, such as the use of transactions and locking, which is typically present in more generic databases.

- Dact [20] follows the exact opposite approach – it stores Alpino dependency structures as-is in a Berkeley DB XML database. Although the use of an XML database incurs some overhead, it makes the data queryable (XPath and XQuery) and processable (XSLT and XQuery) using W3C web standards. As a result, Dact can leverage XML technology extensively. It uses

[3] A more in-depth description of the linguistic annotation can be found in the TüPP-D/Z stylebook [15], and information about the actual XML encoding of linguistic annotation can be found in the TüPP-D/Z markup guide [18].

XPath (with support for macros) as its query language and heavily on XSLT for rendering and data export.

TüNDRA takes the latter route and uses BaseX [6] as its database backend. BaseX is a light-weight XML database that uses XQuery as its query language. To execute a query, TüNDRA's query processor first parses a TIGERSearch query into an intermediate representation, TIQR, that is amendable for query optimization [11]. The TIQR representation is then used to write the XQuery program that is executed by BaseX.

4.2 Motivation for Improving Scalability

TüNDRA relied on a couple of different techniques to make query processing performant. The BaseX database performs indexed queries on attributes of the elements that represent syntax tree nodes to restrict the set of nodes to be analyzed. Moreover, TIQR graph is processed such that attribute values that are infrequent are selected in XQuery before frequent attributes. Despite such optimizations, searching a treebank in TüNDRA could be slow. For example, consider the following query to select nodes (*d*) that dominate an *NX* immediately followed by *PX*:

(5) `#nx:[cat="NX"] . #px:[cat="PX"] & #d > #nx & #d > #px`

Processing such a query is relatively slow, because the (indexed) attributes select for a substantial number of trees (e.g. 75.3% of the sentences in TüBa-D/Z). Moreover, since two categories are considered to be adjacent in the TIGERSearch query language when their lexical nodes are adjacent, the query requires more structural matching than it may seem on the surface. While such a query takes tens of seconds on a treebank such as TüBa-D/Z (95000 sentences), under the assumption of linear scaling it would take hours to process on the TüPP-D/Z (11.5M sentences).

Even if long query processing times are a given, optimization of the user experience alleviates most of that problem. In our redesign of the TüNDRA backend, two principles guided this optimization: (1) the user should see the first query results within seconds. This is motivated by the observation that query formulation is typically an iterative process — a query is refined until it reflects exactly the phenomenon that a user is interested. If the time to the first result is to long, it interferes with this iterative refinement. (2) The user should be able to get intermediate statistics when the query is running. For many queries, one can already get a rough idea of the distribution of results when a fraction of a large treebank is processed. This allows the user to see if there are any interesting trends.

4.3 Architecture

As discussed in the previous sections, attribute indices form the backbone of speedy query processing. For this reason, XML databases generally load or map the indexes into memory in order to process the query. It turns out that for large treebanks, this is the largest impediment to return results as early as possible [3], since the indices get paged out regularly. We solve this problem in TüNDRA by splitting the treebank in chunks that are small enough to make this loading time negligible for each chunk. To present the treebank as one single unit, each chunk is programmatically wrapped in a *multi-treebank*. This multi-treebank does the necessary translations to make it appear as a single treebank, such as: presenting iterators over results in all chunks, rewriting tree identifiers to monotonously increase, extracting/caching treebank metadata, and assuring that each chunk is of the same treebank type. Since our current implementation of the multi-treebank processes chunks sequentially, the mean time to the first match is roughly $E = \frac{t_c}{p_q}$ where t_c is the time to process a chunk and p_q the probability that a hit is found in a chunk for query q.

Another way TüNDRA provides immediate feedback to the user is by providing live query statistics. For instance, if the user executes the query of the previous section, they can view the distribution of the values that occurred for a particular attribute (for instance, *cat*). The statistics window is updated by executing the query asynchronously and updating the statistics window every n seconds. Unfortunately, we found that gathering statistics on large treebanks often resulted in copious memory use, since some queries can result in many distinct values.

For queries that result in an extremely large number of hits, we switch to reservoir sampling [19]. Reservoir sampling is an algorithm that is strongly related to Fisher-Yates shuffling for choosing k out of n items uniformly, where n is unknown beforehand. At each moment, the sample should be representative of query hits *thus far*, assuming that query match values are uniformly distributed across the corpus.[4] The statistics are updated when a match is replaced in the reservoir — the count for the replacee is decreased and that of the replacement increased.

5 Corpus Results on the Zwischenstellung

Table 1 summarizes the corpus results for the Zwischenstellung found in the TüPP-D/Z treebank. With a total of 92 occurrences in a corpus of 11,512,293 sentences, this phenomenon is, indeed, rare and hence requires large corpus resources of the kind used in the present study. The VVINF, VVPP, and VMINF part-of-speech tags in Table 1 are taken from the STTS tagset [16] for German and stand for main verb infinitive, main verb past participle, and modal auxiliary verb infinitive, respectively. While most of the corpus examples involve *können* and *müssen*, they also appear in the TüPP-D/Z with *sollen*, *wollen*, *dürfen*, and *mögen* the other four modal verbs subsumed under the part-of-speech tag VMINF.

The Zwischenstellung is often characterized as dialectal, especially attributed to southern varieties of German, and sometimes as archaic. Interestingly, the cor-

[4] This is a weakness in our current implementation. One possible solution is to shuffle the sentences before use.

pus findings in Table 1 do not confirm either of these claims. With more than 90 occurrences, the Zwischenstellung is well-attested in the TüPP-D/Z treebank. The regional character of the Zwischenstellung is also not confirmed by the TüPP-D/Z. The taz newspaper used for the TüPP-D/Z treebank is published in Berlin, and the particular local taz issue used for the treebank is the Bremen taz edition. While it is not a foregone conclusion that the journalists are from this northern area only, it is highly unlikely that they are all speakers of southern varieties of German.

Linguistic Pattern	Avg. occurrences per 1 million tokens	Raw Corpus frequencies
VVINF ***haben*** VMINF	0.07	15
VVINF ***werden*** VMINF	0.15	30
VVINF ***haben*** VMINF	0.02	4
VVINF ***werden*** VMINF	0.15	26
VVINF ***haben*** *lassen*	0.05	11
VVINF ***werden*** *lassen*	0.01	1
VVPP ***haben*** werden VMINF	0.03	5

Table 1: Zwischenstellung of *haben* and *werden*

The examples in (6) are taken from the TüPP-D/Z treebank. They illustrate each of the seven linguistic pattern listed in Table 1.

(6) a. daß er von Wahlfälschungen nichts wissen habe können.
that he of election fraud nothing known has been able to
'that could know anything about election fraud.'

b. wegen dem der Strauß 62 gehen hat müssen.
because of which the Strauß 62 leave has had to
'because of which Mr. Strauss had to leave in 1962.'

c. ob die sich in der neuen Hochblüte des Kapitalismus
whether they self in the new hayday of capitalism
halten werden können.
keep will be able to
'whether they will be able to persist in this new hayday of capitalism'

d. daß sie so lange Haftstrafen absitzen werden müssen.
that they such long prison terms serve will have to.
'that they will have to serve such long prion terms.'

e. die man laufen hat lassen.
which one walk has let
'which one has let go.'

f. die . . . uns lange vor unserer Hybris erzittern werden
which . . . us long due to our arrogance tremble will
lassen.
let
'which will let us tremble on account of our arrogance.'

g. deren Zustimmung eingeholt hätte werden müssen.
whose consent sought had [Passive werden] have to
'whose consent would have to have been sought.'

Interestingly, the Zwischenstellung occurs also among the 4-element verbal clusters. One possible language-processing explanation for this finding may be that the Zwischenstellung offers an effective way to separate the full verb from the other (auxiliary) verb members of the verb cluster. For the 4-element verbal clusters, with three auxiliaries following the main verb, this clear separation may well facilitate language comprehension and production.

The TüNDRA search queries used to extract instances of the Zwischenstellung from the TüPP-D/Z require reference to the syntactic annotation of the treebank, in particular to the layer of topological field annotation, and reference to the layer of morpho-syntactic part-of-speech annotation.

(7) a. [cat="VCRAFVI"] > #1:[pos = "VVINF"] & #1 . #2: [pos="VAFIN"
& lemma = /haben|werden/] & #2 . #3:[pos="VMINF"
& lemma=/müssen|können|dürfen|wollen|sollen|mögen/]

b. [cat="VCRAFVI"] > #1:[pos = "VVINF"] & #1 . #2: [pos="VAFIN"
& lemma = /haben|werden/] & #2 . #3:[pos="VVINF"
& lemma=/lassen/]

The first terms in the two TüNDRA search queries in (7) use the topological field label VCRAFVI (short for: right-bracket verbal complex (VCR_) with finite auxiliary (_AF) and infinite verb (_VI) and, thus, suitably restrict the search to subordinate clauses. The > operator stands for immediate dominance, and the dot operator (.) for immediate precedence.

Simpler queries that search for sequences of part-of-speech labels and lemmas and that do not include topological field information, as in (8), lack the required accuracy.

(8) #1:[pos="VVINF"] . #2:[lemma=/haben|werden/]
& #2 . #3:[lemma=/müssen|können|dürfen|wollen|sollen|mögen/lassen]

They retrieve as false positives sentences as in (9), where the sequence of matching lexical tokens for query (8) are identified in (9) by corresponding numerical subscripts. The lexical tokens #1 and #2 matching the query do not belong to a single topological field, as is required by query (7a), but straddle the left bracket (VCL) of a main clause and the Vorfeld (VF) and left bracket of a main clause in (9a); hence they do not constitute examples of the Zwischenstellung of the finite auxiliary *wird*. (9b) is not admitted by query (7a), since the auxiliary *kann* does not match the part-of-speech tag VMINF.

(9) a. Einziehen$_1$ wird$_2$ dürfen$_3$, wer dringend ein Dach
move in will be allowed who urgently a roof over
über dem Kopf braucht.
the head needs.
'They will be allowed to move who urgently need a place to stay.'

b. Welche Schlüsse Milosevic daraus ziehen$_1$ wird$_2$ kann$_3$
which conclusions Milosevic from that draw will can
noch niemand voraussagen.
so far nobody predict.
'Which conclusions Milosevic will draw from that so far nobody can predict.'

While TüNDRA search query (7a), which includes topological field information, is highly accurate, it does not succeed in retrieving all cases of the Zwischenstellung construction contained in the TüPP-D/Z treebank. This is due to annotation errors in the treebank data that arise from automatic annotation of the data source. Such annotation errors often orginate at the level of part-of-tagging. For the construction at hand, auxiliaries such as *haben* and *können*, where the finite and non-finite forms coincide, are often mistagged. In order to retrieve examples of the Zwischenstellung for which finite auxiliaries have been mistagged as non-finite (VAINF), the queries in (10) are necessary.

(10) a. [cat="VCRVI"] > #1:[pos="VVINF"] &
#1 . #2:[pos="VAINF" & lemma = /werden|haben/] &
#2 . #3:[lemma=/müssen|können|dürfen|wollen|sollen|mögen/]

b. [cat="VCRVI"] > #1:[pos="VVINF"] &
#1 . #2:[pos="VAINF" & word=/haben|werden/] &
#2 . #3:[lemma=/lassen/]

These queries refer to the same topological field of a right-bracket verbal complex (VCR) that is also included in queries (7). But the queries (10) use a different suffix (_VI) for this topological field since the field contains only non-finite verbs.

While queries (10) lead to the required recall for examples of the Zwischenstellung with mistagged POS tags, they lack precision since they admit a number of false positives. The subscripts in (11) match the hash tags in query (10).

(11) Es wird bereits eine Denkpause gefordert, wie mit den
it werden passive already a moratorium demanded, how with the
Unterlagen der Staatssicherheit weiter verfahren$_1$ werden$_2$
documents of the secret police further proceeded werden passive
soll$_3$.
should .

> 'A moratorium has already been demanded how best to proceed with the documents of the secret police.'

Such false positives include examples such as (11), where the participial and infinitival forms of main verbs coincide, as is the case for the verb *verfahren*. (11) is actually an example of an impersonal passive with *werden* as a passive marker, rather than an instance of *werden* in Zwischenstellung.

In sum, the TüNDRA queries used to extract instances of the Zwischenstellung from the TüPP-D/Z treebank need to be hand-filtered since they are inevitably noisy. This noisiness is due to two main factors: (i) annotation errors in the treebank data that arise from automatic annotation of the data source, and (ii) the imprecision of the queries themselves, which also yield instances of other syntactic constructions, in particular of passive sentences. The manual filtering of such false positives is greatly facilitated by the incremental presentation of query results in TüNDRA described in Section 4 above.

6 Conclusion and Outlook

This paper has reported on the use of the treebank query tool TüNDRA, which is able to process large treebanks of the size needed for rare syntactic constructions such as the Zwischenstellung of finite auxiliaries in German subordinate clauses. The corpus findings about the Zwischenstellung shed new light on the usage of this construction in contemporary German that contradict previous claims contained in the linguistics literature.

The TüPP-D/Z, an automatically annotated treebank with 11.5M sentences, contains a total of 92 examples of this construction, which need to be handfiltered from corpus queries that produce a significant number of false positives. The noisiness of automatically annotated data, incidentally looking at the Zwischenstellung as one of the syntactic constructions under consideration, is addressed also in some detail in [14], whose observations and conclusions are largely confirmed by the present corpus study.

At present, the burden is on the users of the TüNDRA tool to overcome noisiness of annotation by refining their queries in the appropriate way. In future work, it would be interesting to explore to what extent such query refinements can be guided by the tool itself.

7 Acknowlegdements

The authors gratefully acknowledge the financial support of their research by the German Ministry for Education and Research (BMBF) as part of the CLARIN-D research infrastructure grant given to the University of Tübingen.

References

[1] John Ole Askedal. "Ersatzinfinitiv/Partizipialsatz" und Verwandtes: Zum Aufbau des verbalen Schlussfeldes in der modernen deutschen Standardsprache. *Zeitschrift für germanistische Linguistik*, 19, 1–23, 1991.

[2] Gunnar Bech. *Studien über das deutsche verbum infinitum.* Dan. Hist. Filol. Medd. Bind 35, no. 2 (1955) & Bind 36, no. 6 (1957). Det Kongeliege Danske Videnskabers Selskab, 1955/57.

[3] Daniël de Kok, Dörte de Kok, and Marie Hinrichs. Build your own treebank. In *Proceedings of the CLARIN Annual Conference*, Volume 2014, 2014.

[4] Hans den Besten and J. Edmondson. The verbal complex in continental West Germanic. In Werner Abraham, editor, *On the Formal Syntax of the Westgermania. Papers from the '3rd Groningen Grammar Talks'*. John Benjamins Publishing Company, Amsterdam/Philadelphia, 155–216, 1983.

[5] Peter Eisenberg, G. Smith, and O. Teuber. Ersatzinfinitiv und Oberfeld – ein großes Rätsel der deutschen Syntax. *Deutsche Sprache*, 29(1):242–260, 2001.

[6] Christian Grün, Alexander Holupirek, and Marc H Scholl. Visually exploring and querying XML with BaseX. In *BTW*, 103, 629–632, 2007.

[7] John Evert Härd. *Studien zur Struktur mehrgliedriger deutscher Nebensatzprädikate. Diachronie und Synchronie*. Number 21 in Göteborger Germanistische Forschungen. Göteborg, 1981.

[8] Wolfgang Lezius. TIGERSearch ein Suchwerkzeug für Baumbanken. *Tagungsband zur Konvens*, 2002.

[9] Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948, 1993.

[10] Mitchell P. Marcus, Beatrice Santorini and Mary Ann Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 313–330, 1993.

[11] Scott Martens. Tündra: Tigersearch-style treebank querying as an xquery-based web service. In *Proceedings of the joint CLARIN-D/DARIAH Workshop "Service-oriented Architectures (SOAs) for the Humanities: Solutions and Impacts"(DH 2012)*, 41–50, 2012.

[12] Scott Martens. TüNDRA: A web application for treebank search and visualization. In *The Twelfth Workshop on Treebanks and Linguistic Theories (TLT12)*, 133–144, 2013.

[13] Paul Meurer, Miriam Butt, and Tracy Holloway King. Iness-search: A search system for LFG (and other) treebanks. In *Proceedings of the LFG'12 Conference*, 404–421, 2012.

[14] Walt Detmar Meurers. On the use of electronic corpora for theoretical linguistics. Case studies from the syntax of German. *Lingua*, 115 (11), 1619–1639, 2005.

[15] Frank Henrik Müller. Stylebook for the Tübingen Partially Parsed Corpus of Written German (TüPP-D/Z). Technical report, University of Tübingen, Seminar für Sprachwissenschaft, 2004.

[16] Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. Guidelines für das Tagging deutscher Textcorpora mit STTS (kleines und großes Tagset). Technical report, Universität Stuttgart, Universität Tübingen, Tübingen, Germany, 1999.

[17] Heike Telljohann, Erhard Hinrichs, Heike Zinsmeister, and Kathrin Beck. Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Technical report, University of Tübingen, Seminar für Sprachwissenschaft, 2015.

[18] Tylman Ule. Markup Manual for the Tübingen Partially Parsed Corpus of Written German (tüpp-d/z). Technical report, University of Tübingen, Seminar für Sprachwissenschaft, 2004.

[19] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

[20] Gertjan Van Noord, Gosse Bouma, Frank Van Eynde, Daniel De Kok, Jelmer Van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste. Large scale syntactic annotation of written Dutch: Lassy. In *Essential Speech and Language Technology for Dutch*, 147–164. Springer, 2013.

Taking Care of Orphans: Ellipsis in Dependency and Constituency-Based Treebanks

Tomáš Jelínek, Vladimír Petkevič, Alexandr Rosen,
Hana Skoumalová and Přemysl Vítovec

Institute of Theoretical and Computational Linguistics
Faculty of Arts, Charles University in Prague
E-mail: `firstname.lastname@ff.cuni.cz`,
`premysl.vitovec@gmail.com`

Abstract

Using results of dependency parsers, where a syntactic daughter missing its parent is dependent on its grandparent, we build a constituency-based treebank, where such cases of ellipsis are restored as appropriately headed phrases, as long as no other solution is available. We arrive at structures of three types: (i) non-elliptical (as in pro-drop ellipsis), (ii) elliptical with the head and mother nodes restored, (iii) elliptical with the original dependency labelling retained. Finally, we evaluate the processing steps – morphological annotation, dependency parsing, conversion from dependency to constituency, and expressing ellipsis in the resulting structure licensed by a formal grammar.

1 Introduction

Ellipsis is a challenging phenomenon for any linguistic theory, formalism or application. We focus on the issue of detecting and representing elliptical constructions in the process of building a constituency-based treebank of Czech from the output of stochastic dependency parsers (Jelínek et al. [9], Petkevič et al. [19]). The texts come from the SYN2015 corpus,[1] consisting of 100 million tokens in fiction, professional and newspaper texts.

The resulting annotation should be (i) consistent with a formal grammar-like specification; (ii) produced by software tools only to allow for annotating large data; (iii) offering different representations of syntactic structure to satisfy users with different preferences and/or theoretical backgrounds, ranging from linguistic experts to teachers and students even at the secondary school level.[2]

[1] See http://korpus.cz.

[2] Although the users' preferences have not been elicited in a formal way, at least some of them

Due to the lack of manually annotated constituency treebank, the input to the treebank building procedure is a dependency parse in the format of the analytical level (a-level) of the Prague Dependency Treebank – PDT (Hajič et al. [7], PDT [18]), where ellipsis is not represented explicitly. This is due to the 1:1 relationship between text tokens and nodes of the dependency tree (except for the root node), which leaves no space for any restored nodes. It is the more abstract, tectogrammatical level (t-level) of PDT, where ellipsis is represented consistently.[3] However, t-level does not seem to be a better candidate for our purpose. In addition to a more error-prone procedure needed to arrive at the more abstract structure and to add information related to ellipsis, the t-level is also too abstract for our task, mainly because function words are represented only as features of content words.

Several clues play a part in the task of detecting ellipsis in the a-level-style parse: analytic (surface) syntactic function (afun), structure and word class. The most useful clue is the ExD label, standing for the "external dependent" afun. Nodes labeled as ExD often represent remnants in constructions which are assumed to be elliptical, while the missing parts, subject to ellipsis, are restored on the t-level. These "orphaned" ExD remnants are immediate dependents on their grandparent nodes (or even more distant ancestors), often sisters to regular children of the grandparent, with some additional complexities when coordination or function words are involved. Since their afun is specified as ExD, their function in the elliptical construction is not identified. As a result, their position in the syntactic tree and their afun label are both counter-intuitive and at odds with a constituency-based approach.

At the present stage, antecedents of elided parts are not identified. Instead, we rely on "standard" rules of syntax to specify as much as possible about the missing words or constituents, using the clues available in the dependency parse together with external lexical information and constraints in the grammar to identify the syntactic function of the orphan with respect to its head and the grammatical categories of the head from the local context.

In §2 we provide a brief overview of some theoretical accounts of ellipsis relevant to our topic. Next in §3 we describe how ellipsis can be detected in the parse structured as the a-level of the PDT, propose a corresponding representation in a constituency-based treebank and show how the structures are converted. We proceed with §4, concerned with the practical issues of processing elliptical constructions by the toolchain yielding the constituency-based treebank. Most of this section is concerned with the evaluation of the steps of this procedure. Finally, §5 summarizes the result.

can be approximated given the authors' experience with the users' feedback for the Czech National Corpus and an existing Czech dependency-based treebank.

[3] Although both a-level and t-level are inspired by the theory of Functional Generative Description (FGD, Sgall et al. [22]), it is the t-level which is a more faithful incarnation of its assumptions.

2 Ellipsis in linguistic theory

The challenging status of ellipsis is reflected in the fact that there is no consensus about its typology or even about the range of covered phenomena. It is due mainly to the fact that ellipsis often defies the standard notion of a constituent and its interpretation hinges on the interplay of morphology, syntax, semantics and information structure. The definition is often phrased in a very general way: "The term [...] refers to the omission of linguistic material" (Brown, [5, p. 109]). Within mainstream generative linguistics (Merchant [14], Kayne [11], Kosta [12]) at least some of the following eight types of ellipsis are distinguished:[4] **(i)** GAPPING: the gap includes a finite verb – *Kim* ***can play*** *the piano and Alex _ the guitar*; **(ii)** STRIPPING: gapping with a single remnant – *Kim* ***can play*** *the piano and Alex _, too*; **(iii) VP-**ELLIPSIS: of a non-finite VP – *Kim can* ***play the piano*** *but Alex can't _*; **(iv)** PSEUDOGAPPING: the remnant includes an auxilliary and a partial VP – *Kim* ***plays*** *the piano better than Alex does _ the guitar*; **(v)** SLUICING: a bare wh-phrase remains – *Kim* ***can play*** *something but I don't know what _*; **(vi)** NOUN ELLIPSIS: a noun is omitted, possibly with some modifiers – *Kim plays Mozart's* ***etudes*** *and Alex plays Chopin's _*; **(vii)** ANSWER FRAGMENTS: answer omits redundant information present in the question – (*Who* ***can play the piano?***) *Kim _*; and **(viii)** COMPARATIVE DELETION – *Sandy* ***plays the guitar*** *better than Alex _*. All of these types and some additional ones are common also in Czech, many of them involving ellipsis of heads.

The standard approach of transformational theory assumes that a constituent including ellipsis is deleted after the remnant part is moved out of it. A different proposal treats ellipsis as a kind of anaphor, present in the underlying representation. Some proposals couched in alternative generativist theories, such as Combinatory Categorial Grammar (CCG) or Head-driven Phrase Structure Grammar (HPSG), argue that remnant strings such as *Alex the guitar* in (i) above should be treated as a single non-standard constituent (Steedman [23], Mouret [16], Levine [13]). However, there is a substantial support for deletion-based analysis even in the constraint-based theory camp, the framework behind our treebank annotation model (Beavers & Sag [1], Yatabe [26]). Deletion seems to best answer our concern about interpretability of the treebank annotation according to user-specific preferences: while only the remnant parts of an elliptical construction are present as structural elements (tree leafs), the entire construction, including ellipsis, can be restored within an appropriate phrasal node dominating the remnant parts.

There is no generally accepted definition or typology of ellipsis in the (more traditional) Czech linguistic tradition either, see Karlík [10, p. 122–123] for an overview. According to Daneš at al. [6], only a small set of omitted strings count as ellipsis while the antecedent must be unambiguously identifiable. On the other hand, for Mikulová [15] the set of elliptical phenomena is much wider, e.g. including cases of "systemic (grammatical) ellipsis" *pro* and *PRO* as null subject. It is

[4] The list is not exhaustive, some less common types are omitted.

this definition and typology that – albeit in a modified form – underlies the PDT approach to ellipsis.

3 Dependency and constituency-based accounts of ellipsis

Together with Xia et al. [25] we believe that the next-generation treebank should be multi-representational, with parallel annotation layers, or with various representation options of a single annotation. To support this goal, our core annotation represents syntactic structure as constituency-based trees, which typically include more information than dependency-based structures. On the other hand, (internally unstructured) constituents can be used to model underspecified representation, including ambiguous or partial parses. Moreover, the distinction of syntactic mothers from head daughters in constituency trees helps to capture some syntactic phenomena, including ellipsis, in a more natural way. More specifically, constituents can accommodate information about elided daughters even if the daughters are not represented as nodes in the tree.

However, we start from a dependency parse, the result of a combination of the best-performing dependency parsers, trained on the PDT a-level data.[5] The output is converted to a skeletal constituency format, which is checked and populated with additional information, drawn from a lexicon and grammar. The conversion proceeds fully automatically: on each level of the input tree, every bundle, i.e. an elementary dependency tree consisting of a governor and its immediately dependent nodes, is converted to a constituent, consisting of a mother node and its daughters, one of which is typically a head daughter sharing some values with its mother.

Although our input is in the PDT a-level format, it is useful to have a closer look at the more abstract t-level, where both grammatical (systemic) and textual ellipsis are restored, while their types and antecedents are identified. This is not true about the a-level (Mikulová [15, p. 115nn.]). However, the two levels are interlinked: an element on the t-level is linked to a corresponding element on the a-level. An elided content word, omitted on the a-level, is restored on the t-level. If its lexical meaning or other properties can be inferred from the a-level, the corresponding a-nodes are linked with the restored t-node. For instance, in the t-level annotation of *John would go to the cinema, Paul _ to the theatre* the second clause receives a complex node representing the elided verb form (*would go*), linked to its t-level antecedent and to the same two a-level nodes as the antecedent: the content verb *go* and the auxiliary *would*.

As a part of a solution to some types of textual ellipsis, a-level uses the ExD function to label a node whose immediate governor is missing or it is used as an auxiliary symbol if the tree could not be built otherwise. The ExD node depends on the nearest ancestor of its missing governor. On the other hand, an elided leaf

[5] The PDT data are used only for training the parsers. The texts included in the SYN2015 that we annotate are not part of PDT.

is absent without trace on the a-level and is not restored in the target constituency treebank either. There are three exceptions to the "nearest ancestor" rule: (i) ExD is not assigned to a preposition but rather to its dependent noun; (ii) ExD is not assigned to a complementizer but rather to its immediate dependent; (iii) in coordination and apposition, ExD is not assigned to the governing conjunction or punctuation sign but rather to all conjuncts or apposition members. In all the three types of structures the governor is assigned the same function as in non-elliptical structures.

ExD is also assigned to nodes in sentences lacking a finite verb, under the assumption that the verb is elided. This concerns noun phrases such as *Lidové noviny* (a newspaper title) or *Lékaře!* 'Doctor$_{Acc}$!' or other single-word sentences such as *Ano!* 'Yes!'.

In the PDT data, ExD is not very frequent (3.4% of the total number of tokens including punctuation), but it appears in more than 1/4 (26.4%) of sentences, with 2.2 ExDs on average in such a sentence. Mean length of sentences including ExD is 14.4 tokens, shorter than the average sentence in PDT (17.1 tokens), mainly due to the presence of ExD in short verbless sentences.

The target format is a treebank where constituents annotated as typed feature structures with value sharing. The structures are licensed by a modified HPSG signature and constraints on the feature values, a "treebank grammar". The phrasal skeleton, morphosyntactic categories of the lexical items and syntactic functions are derived from the dependency parse. After the parse is matched with lexically-specific information from a valency lexicon (if the relevant entries are available), the grammar constraints make sure that the lexical information is projected to the phrasal categories and the valency requirements are saturated.

The treatment of elliptical constructions is motivated by a few general considerations: (i) Nodes are added only if the tree cannot be built without them. We assume ellipsis only where it is difficult or impossible to treat the structure as grammatical. (ii) All potentially elliptical structures interpretable as regular phrases are treated as such. (iii) Irregular structures where ellipsis cannot be identified reliably are treated as fragments. (iv) An omitted element can but does not have to be restored as a tree node. Its absence in the elliptical structure is specified as the value of the GAP feature of the dominating phrasal category.[6] (v) The GAP value is shared with the head or a complement specification of the dominating category. (vi) A construction with the head omitted is represented by a phrasal category whose head daughter can be missing.

Thus, we annotate ellipsis more sparingly than the t-level of PDT. Our approach is closer to Daneš et al. [6] and the Penn Treebank (Taylor et al. [24], Bies et al. [2]). In the latter, sentences consisting of a single NP or VP are annotated as such, the remnants in a parallel construction are linked to their counterparts in

[6] We are aware of the fact that the GAP feature has been used in the context of HPSG for a different kind of phenomenon (Sag et al. [21]). We hope to find a better name before the first public release of the treebank is due.

the preceding “pattern” as in ((NP-1 *Mary* (VP *likes* (NP-2 *Bach*))) *and* ((NP=1 *Susan*) (NP=2 *Beethoven*))), and simple symbols are used for other types of missing elements (traces, null subjects, omitted complementizers). On the other hand, in our approach the elided elements can be restored in the annotation of the mother node, representing the elliptical construction, and make the elliptical construction “complete” within the larger syntactic context.

Constructions including a word labelled ExD on the PDT a-level (Hajič et al. [7, § 3.4]) are converted in three possible ways: (a) as a non-elliptical structure, (b) as a structure preserving the ellipsis, (c) as a structure including an ExD element, which is a temporary failsoft measure in cases no other solution is available.[7] No ellipsis is assumed in cases where a clause consists of a single NP, VP etc. (1) or in comparison (2).

(1) *Lékaře!*
doctor.ACC
‘Doctor!’

(2) *Bohouš je zdravý jako ryba*
Bohouš is healthy as fish
‘Bohouš is healthy as a fish.’

In other detectable cases, ellipsis is preserved. The ExD orphan node is adopted by a new phrasal mother, the missing head is restored as the value of the mother’s GAP feature.[8] If possible, the orphan’s syntactic function relative to its head is identified from the syntactic and morphological context. The cases include the ellipsis of (i) a predicate in repetion (3), a copula (4) – both examples of gapping; (ii) parts of an analytical verb form – VP-ellipsis (5); (iii) a participle (6); and (iv) a noun and a predicate in repetition (7).

(3) *Kristýna přinesla růži, Jiří fialky.*
Kristýna brought rose Jiří violets.
‘Kristýna brought a rose, Jiří violets.’

(4) *Pátrání zastaveno.*
Investigation stopped.
‘The investigation [has been] suspended.’

(5) *Doufal jsem, že budeme malovat, ale nebudeme.*
hoped.PTCP.SG be.PRS.1SG that be.FUT.1PL paint.INF but be.FUT.1PL.NEG
‘I hoped that we will decorate the flat but we won’t.’

(6) *Jan vstoupil hlavu sklopenou.*
Jan entered head.ACC bent.
‘Jan entered his head down.’

[7] In some cases, ellipsis is detected and represented even when no ExD element is present, e.g. when a preposition is not followed by an NP.

[8] Optionally also as its head daughter, i.e. as a tree node.

(7) *Honza má červený svetr a Eva zelený.*
Honza has red jumper and Eva green.
'Honza has a red jumper, Eva a green one.'

Fig. 1 shows the input dependency structure for (7), Fig. 2 shows the same sentence after conversion to constituency tree.[9] In Fig. 2 the second conjunct (membCo) includes two instances of ellipsis: the verb *má* and the noun *svetr* (*Eva* ***má*** *zelený* ***svetr***). The "head" sister of the subject in the second conjunct clause represents the predicate. Its +GAP specification means that the head of the predicate ***má*** *zelený* ***svetr***, i.e. the verb *má*, is missing. The "obj" node stands for the object constituent *zelený* ***svetr***. Here the +GAP feature means that the noun head of the object constituent is missing.[10] The full clause without ellipsis is shown in Fig. 3.

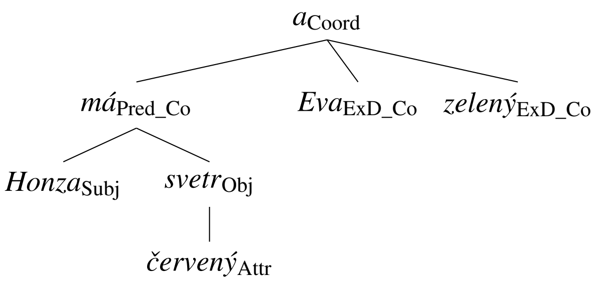


Figure 1: The input dependency structure of (7)

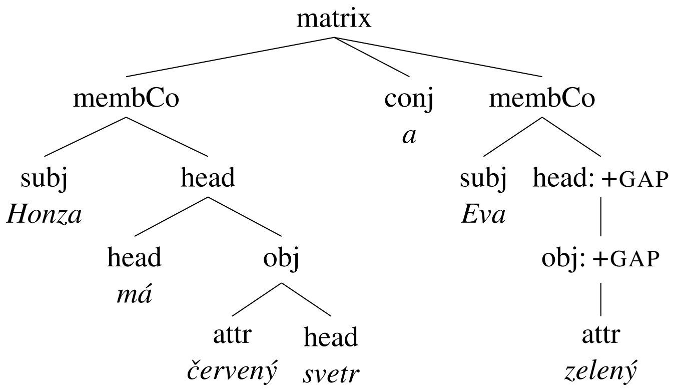


Figure 2: Example (7) after conversion to constituency structure, with restored mother nodes in the clause *Eva _ zelený _* 'Eva a green one'

[9] To match the dependency-based annotation, the node labels stand for syntactic functions rather than categories.

[10] In the full analysis, the GAP values are underspecified representation of the missing elements, derived from available information. E.g. the missing verb is known to be a finite 3rd person singular feminine form, given its subject *Eva*.

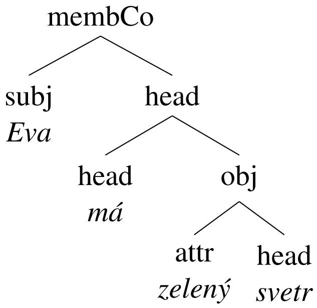


Figure 3: The part of Fig. 2 including ellipsis, with the gaps filled in: *Eva má zelený svetr* 'Eva has as a green jumper'

We do not attempt to restore entire analytical verb forms (8). The example would have the same structure for the second conjunct as in Fig. 2.

(8) *Honza by si byl koupil červený svetr a Eva zelený.*
Honza be.COND self.DAT be.PTCP buy.PTCP red jumper and Eva green
'Honza would have bought a red jumper and Eva a green one.'

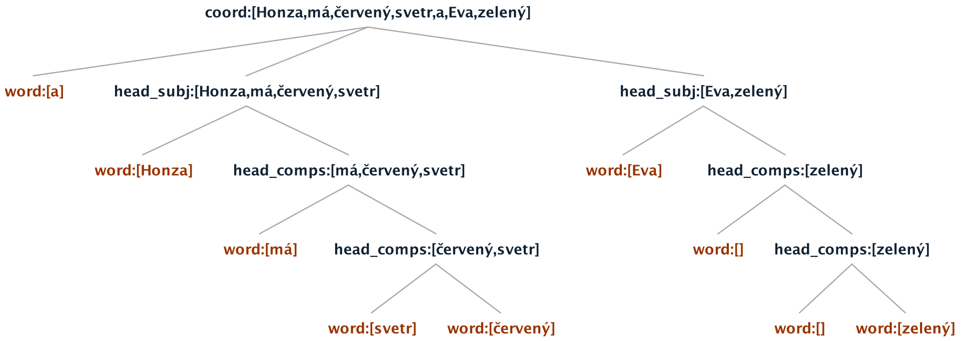


Figure 4: The structure for (7) after it is checked by the grammar

Fig. 4 shows the structure for (7) after it is checked by the grammar.[11] Here the node labels consist of the constituent type ("word" for the terminal node or other label for a subtype of phrase), followed by a string of word tokens corresponding to the constituent. The elided tokens are represented as nodes with empty strings.

The structure of the predicate (VP) part of the second conjunct is shown in Fig. 5 together with the nodes annotated by typed feature structures. The labels in italics stand for atomic values or types of structures. Attributes of the structures (feature names) are in small capitals. The boxed numbers indicate identity of

[11] The graphical display of the output is due to Martin Lazarov's GraleJ package, see https://code.google.com/p/gralej/.

feature values across the structure. Angle brackets denote lists. Syntactic daughters are shown as tree nodes for reading convenience, but internally they are embedded within the feature structure of their mother as values of attributes such as HEAD_DTR or COMP_DTRS.

The value of the PHON feature is a list (string) of word forms represented by the node. Note that the overt string corresponding to the VP is a single adjective. The topmost head-complements phrase (*head_comps*, the VP) consists of an empty lexical (verbal) head (a *word* with TR_SFUN *head* – the syntactic function as determined by the parser) and another *head_comps* phrase whose TR_SFUN is *obj*. The empty lexical head daughter of the latter phrase is a noun modified by an attribute (*attrPlain*). Each of the two heads projects its head features by sharing the CAT attribute with its mother. The heads also have non-empty valency slots for the relevant "syntactic and semantic" (*ss*) parts of their complements (COMPS) and subjects (SUBJ).[12] The GAP value is the elided head daughter, shown in Figs. 4 and 5 as tree nodes.[13]

The restored structure is based on morphosyntactic, functional and structural information in the input parse and on constraints of the grammar. The specific values of the empty nodes are determined by local constraints on agreement and valency satisfaction. In the absence of any lexical information about the missing heads in the input, their word class (*sFin* for finite verb and *iNoun* for noun) and morphosyntactic categories are due to the declarative grammar constraints. Identification of antecedents for ellipsis is a topic for future research.

4 Evaluation of the treatment of ellipsis step by step

The treebank is built automatically in the following steps: (i) morphosyntactic annotation, (ii) dependency parsing, resulting in the PDT a-level format, (iii) rule-based conversion into phrase structure, (iv) checking and augmenting by a treebank grammar and valency lexicon.

Assuming that the presence of ellipsis affects the performance of all tools in the toolchain, only sentences that supposedly include ellipses were selected for evaluation. Thus the test set consists of 308 sentences from PDT including one or more syntactic functions labelled ExD.

4.1 Morphosyntactic tagging

The sentences were tagged by a hybrid tagger (Hnátková et al. [8]) and the result was compared with the morphosyntactic annotation in PDT. The overall accuracy in terms of tokens with correctly assigned tags was 95.32%, with the individual types of errors summarized in Table 1.

[12] The subject is not shown in the tree for space reasons. The value of the noun head's SUBJ is an empty list, while its COMPS lists a single complement, the attribute. The grammar adopts the

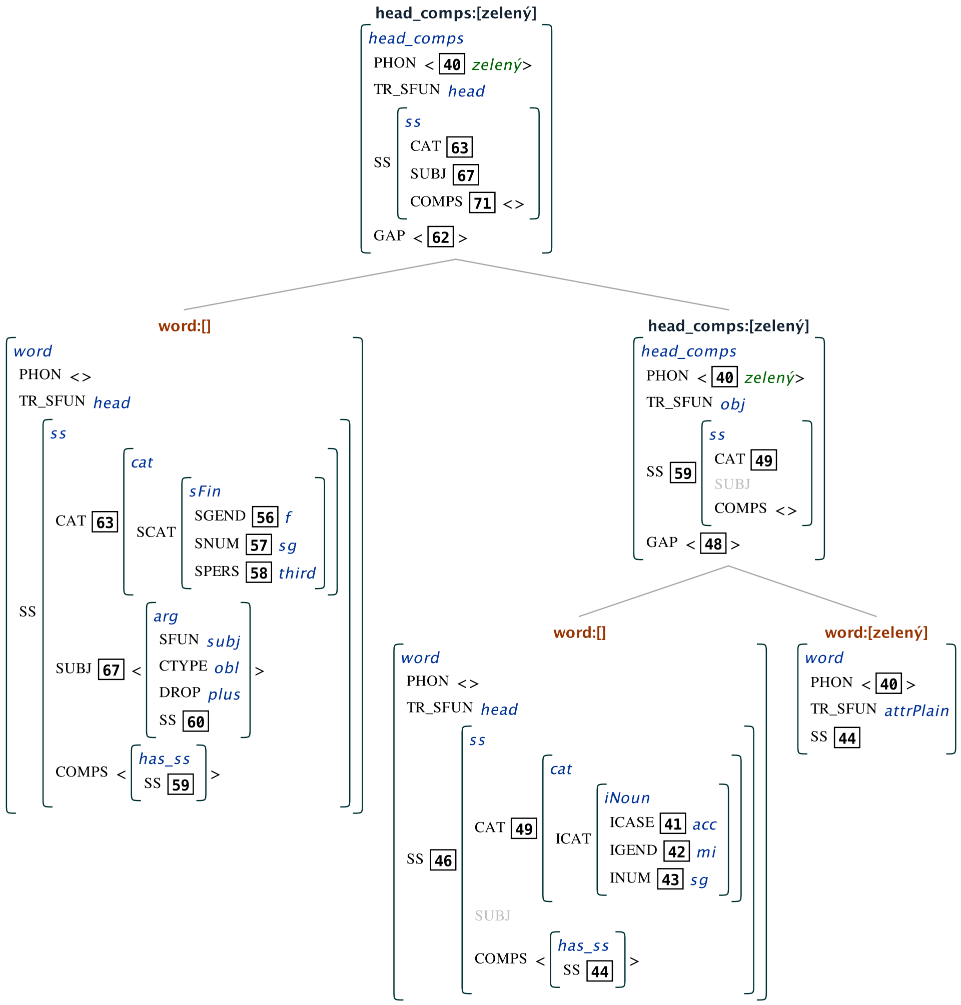


Figure 5: A part of Fig. 4 showing two instances of ellipsis

The first line specifies the error rate with respect to all words in the set of selected sentences, the second one specifies the error rate with respect only to the words assigned the ExD syntactic function in this set:

1. Errors in NPs/PPs (with a noun as the head) concern nouns (1.28/1.92%), adjectives (0.74/0.17%) and prepositions (0.09/0.00%) and they are due to a wrong assignment of case or number (with nouns and adjectives), or of prepositional valency requirements (with prepositions). As the linguistic rules used in the tagger try to identify the whole NPs and PPs, an error in case or number always affects the whole phrase.
2. Errors in pronouns (0.70/0.53%). The errors are mainly caused by a wrong

"adjuncts as complements" approach (Bouma et al. [4], Przepiórkowski [20]).

[13] In the previewed visualization, the user will have the option of displaying or hiding such nodes.

	Noun	Adj	Pron	Num	Verb	Prep	POS	Unkn	Total
all w.	1.28	0.74	0.70	0.16	0.82	0.09	0.30	0.60	**4.69**
ExD w.	1.92	1.17	0.53	0.64	0.21	0	0.96	1.17	**6.62**

Table 1: Breakdown of error rates of morphosyntactic annotation

identification of gender of the pronoun *ten/to* ‘that’ in indirect cases.

3. Errors in verbs (0.82/0.21%) are mainly the result of the wrong assignment of gender in past participles.
4. Errors in numerals (0.16/0.64%) concern mainly errors in gender.

From the viewpoint of the further processing, i.e. syntactic parsing, the most harmful are the errors in POS and in NPs/PPs. Specific disambiguation rules based on a detailed error analysis of the parsed data will have to be developed.

4.2 Parsing

The success of the conversion step relies to a large extent on the successful assignment of the ExD label to the appropriate nodes by the parser. Unfortunately, the assignment of ExD in terms of recall and precision to tokens labelled as ExD in PDT is not very reliable: 66.5/74.2% for the MaltParser (Nivre et al. [17]) and 69.6/78.5% for the Mate parser (Bohnet & Nivre [3]). The assignment of ExD labels was evaluated on the entire set of PDT a-layer data – 87,738 sentences, using ten-fold cross-validation. The rather low precision and recall are due to the relatively low frequency of words labelled as ExD in the data (3.4%), and to the ill-defined and unintuitive guidelines for the assignment of ExD to several phenomena, which are only loosely related.

4.3 Conversion to the constituency format

The same set of 308 sentences as evaluated in §4.1 (primarily newspaper sentences) including a correctly assigned ellipsis in the input data were selected for evaluation of the conversion from dependency to constituency structure; each of them included one or more nodes assigned the ExD (External Dependent) syntactic function. The sentences included 332 instances of ellipsis (ExD), categorized as follows:

1. Ellipsis in input sentence is expressed as a non-elliptical structure on output:
   (a) Omission of a head noun (only adjectival attributes are present as the head noun’s daughters): 15 instances, no error
   (b) In comparative constructions, deletion of:
      i. A noun phrase: 3 instances, no error
      ii. Another syntactic element: 33 instances, 4 errors (12.1%)
   (c) Omission of another type (typically a head verb elided):
   211 instances, 10 wrong (4.15%)
   This type mainly concerns the ellision of a verb in titles and headlines.

2. Ellipsis in an input sentence is converted to a structure preserving the ellipsis: the ExD orphan node is adopted by a new phrasal mother and the missing head is restored as the value of the mother's GAP feature:
   40 instances, 33 wrong (82.5%)
   This type concerns primarily the elements in parentheses as in (9), where the elements in brackets should be treated as non-elliptical, which is a clear error in the conversion program.

   (9) *Přivedla na jednu scénu v hlavních rolích Karla Rodena (Don*
   brought.F.SG.3RD to one stage in leading roles Karel Roden (Don
   Juan) a Miroslava Táborského (Faust)
   Juan) and Miroslav Táborský (Faust)
   'She presented Karel Roden (Don Juan) and Miroslav Táborský (Faust) in leading roles at a single stage.'

3. The output structure preserves the ExD element on input: 30 instances, no error. This is a temporary expedient. We expect that the number will decrease as the refinement of the conversion will proceed.

4.4 Evaluation of the grammar

We tested 308 sentences with ellipsis, which were evaluated in the previous conversion part, with the result of 130 sentences successfully checked by the grammar (error rate: 57.8%). The sentences that did not succeed in the checking procedure either contained several ellipses, or were too complex for our grammar to cope with.

The test set contains 5694 tokens, and the average sentence length is 18.5 tokens. Most sentences (36) have only 3 tokens, the longest sentence has 80 tokens. Most sentences that were successfully analyzed by the grammar have less than 8 tokens in length (89 sentences), the longest checked sentence has 43 tokens. Most sentences that failed are rather long.

4.5 Summary of the evaluation

In a randomly selected set of 308 sentences from PDT, including 332 instances of ExD, the steps of the procedure perform as follows:

1. Morphosyntactic tagging:
   95.32% accuracy
2. Parsing in terms of recall / precision – the only step evaluated on sentences including at least one ExD extracted from the set of PDT a-layer data (87,738 sentences):
   66.5% / 74.2% for MaltParser
   69.6% / 78.5% for Mate parser
3. Conversion from dependency to constituency:
   85.85% of sentences converted correctly

4. Grammar performance:
   42.20% of sentences checked successfully

5 Conclusions

There are several points where a constituency-based account of ellipsis seems to make better sense than the solution adopted on the PDT a-level, including the practical aspect of supporting queries targeting specific types of ellipsis: (i) a more intuitive representation due to a more appropriate position of orphans (the remnant parts of elliptical constructions) in the syntactic structure; (ii) orphans are assigned a syntactic function label according to their role in the elliptical construction; (iii) hypotheses about the missing head can be derived from the local context using standard constraints on agreement and valency; (iv) a more uniform structure for both gapped and gapless structures is available; (v) there is a more direct link to deep syntax or semantics. However, we still base our procedure detecting some cases of ellipsis on the results of parsers trained on the PDT dependency structure with some nodes labelled as ExD. In this sense, the PDT annotation represents an invaluable resource.

In order to improve the treebank annotation, further work will focus on the following three areas: (i) morphosyntactic tagging: more sophisticated disambiguation rules concerning primarily parts of speech will be developed in the rule-based part of the hybrid tagger; (ii) more specific dependency to constituency conversion rules are necessary to eliminate the mere transfer of ExD; (iii) refinement of grammar constraints concerning the missing heads, with the option of identifying antecedents.

Acknowledgment

The work on this paper was supported by grant No. GAČR P406/13-27184S.

References

[1] John Beavers and Ivan A. Sag. Coordinate ellipsis and apparent non-constituent coordination. In Stefan Müller, editor, *Proceedings of the HPSG-2004 Conference, Center for Computational Linguistics, Katholieke Universiteit Leuven*, pages 48–69. CSLI Publications, Stanford, 2004.

[2] Ann Bies, Mark Ferguson, Karen Katz, Robert MacIntyre, Victoria Tredinnick, Grace Kim, Mary Ann Marcinkiewicz, and Britta Schasberger. Bracketing guidelines for Treebank II style Penn Treebank project. Technical report, University of Pennsylvania, 1995.

[3] Bernd Bohnet and Joakim Nivre. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[4] Gosse Bouma, Rob Malouf, and Ivan A. Sag. Satisfying constraints on extraction and adjunction. *Natural Language and Linguistic Theory*, 1(19):1–65, 2001.

[5] Keith Brown, editor. *Encyclopedia of Language & Linguistics*. Elsevier, 2nd edition, 2005.

[6] František Daneš, Miroslav Grepl, and Zdeněk Hlavsa. *Mluvnice češtiny 3 – Skladba [Grammar of Czech 3 – Syntax]*. Academia, Praha, 1987.

[7] Jan Hajič, Jarmila Panevová, Eva Buráňová, Zdeňka Urešová, and Alla Bémová. A Manual for Analytic Layer Annotation of the Prague Dependency Treebank (English translation). Technical report, ÚFAL MFF UK, Prague, Czech Republic, 1999.

[8] Milena Hnátková, Vladimír Petkevič, and Hana Skoumalová. Linguistic Annotation of Corpora in the Czech National Corpus. In Труды международной конференции "Корпусная лингвистика – 2011", pages 15–20. St.-Petersburg State University, Institute of Linguistic Studies (RAS), Russian State Herzen Pedagogical University, 2011. ISBN 978-5-8465-0005-5.

[9] Tomáš Jelínek, Vladimír Petkevič, Alexandr Rosen, Hana Skoumalová, Přemysl Vítovec, and Jiří Znamenáček. A grammar-licensed treebank of Czech. In Verena Henrich, Erhard Hinrichs, Daniel de Kok, Petya Osenova, and Adam Przepiórkowski, editors, *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories (TLT13)*, pages 218–229, Tübingen, 2014.

[10] Petr Karlík, Marek Nekula, and Jana Pleskalová, editors. *Encyklopedický slovník češtiny*. Nakladatelství Lidové noviny, 2002.

[11] Richard S. Kayne. *Movement and Silence*. Oxford Studies in Comparative Syntax. Oxford University Press, Oxford and New York, 2005.

[12] Peter Kosta. *Leere Kategorien in den nordslavischen Sprachen. Zur Analyse leerer Subjekte und Objekte in der Rektions-Bindungs-Theorie*. Frankfurt am Main, 1992.

[13] Robert Levine. Linearization and its discontents. In Stefan Müller, editor, *The Proceedings of the 18th International Conference on Head-Driven Phrase Structure Grammar*, pages 126–146, Stanford, 2011. CSLI Publications.

[14] Jason Merchant. *The Syntax of Silence: Sluicing, Islands, and the Theory of Ellipsis*. Oxford University Press, Oxford and New York, 2001.

[15] Marie Mikulová. *Významová reprezentace elipsy*, volume 7 of *Studies in Computational and Theoretical Linguistics*. Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, Prague, 2011.

[16] François Mouret. A phrase structure approach to argument cluster coordination. In Stefan Müller, editor, *The Proceedings of the 13th International Conference on Head-Driven Phrase Structure Grammar*, pages 247–267, Stanford, 2006. CSLI Publications.

[17] Joakim Nivre, Johan Hall, and Jens Nilsson. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC-2006*, pages 2216–2219, Genova, 2006. ELRA.

[18] Prague Dependency Treebank, 2013. Version 3.0, http://ufal.mff.cuni.cz/pdt3.0/.

[19] Vladimír Petkevič, Alexandr Rosen, and Hana Skoumalová. The grammarian is opening a treebank account. *Prace Filologiczne*, 2015. In print.

[20] Adam Przepiórkowski. On Complements and Adjuncts in Polish. In Robert D. Borsley and Adam Przepiórkowski, editors, *Slavic in HPSG*, Studies in constraint-based lexicalism, pages 183–210. CSLI Publications, Stanford, 1999.

[21] Ivan A. Sag, Thomas Wasow, and Emily M. Bender. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, CA, 2 edition, 2003.

[22] Petr Sgall, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Reidel and Academia, Dordrecht and Praha, 1986. Editor: Jacob Mey.

[23] Mark Steedman. *The Syntactic Process*. MIT Press, Cambridge, MA, 2000.

[24] Ann Taylor, Mitchell Marcus, and Beatrice Santorini. The Penn Treebank: An overview, 2003.

[25] Fei Xia, Owen Rambow, Rajesh Bhatt, Martha Palmer, and Dipti Misra Sharma. Towards a multi-representational treebank. pages 127–133, Utrecht, 2009. LOT.

[26] Shûichi Yatabe. Comparison of the ellipsis-based theory of non-constituent coordination with its alternatives. In Stefan Müller, editor, *Proceedings of the 19th International Conference on Head-Driven Phrase Structure Grammar, Chungnam National University Daejeon*, pages 453–473, 2012.

Integrating Polish LFG with External Morphology

Katarzyna Krasnowska-Kieraś and Agnieszka Patejuk

Institute of Computer Science
Polish Academy of Sciences
E-mail: `kasia.krasnowska@gmail.com, aep@ipipan.waw.pl`

Abstract

This paper presents a previously undocumented approach to combining an extensive LFG grammar, employed in the construction of an LFG parsebank, with an exhaustive external morphological component. It shows how the Polish morphological analyser Morfeusz is plugged into the XLE grammar architecture as a basis for tokenisation and morphological analysis steps. The proposed solution also takes into account phenomena such as the treatment of MWEs and abbreviations. Finally, it is demonstated how the tokeniser and analyser components interact with the grammar rules.

1 Introduction: problem and previous solutions

This paper presents a previously undocumented approach to integrating an exhaustive external morphological component, Morfeusz[1] [15], with an extensive XLE/LFG grammar for Polish, POLFIE [5], in an effective and economic way. The proposed method is general and could be adapted for use by other XLE/LFG grammars.

The grammar is employed in the construction of an LFG treebank of Polish [6], more specifically, a "parsebank" containing both constituency structures and functional structures which feature dependency-like information, among other linguistic information. To obtain valence information, POLFIE uses converted entries from Walenty [9] – a state-of-the-art valence dictionary of Polish.

The previous solution, briefly described in [5], used a Python script to create lexicon files containing the entries for the exact forms (rather than lemmata) found in the sentence to be parsed – such entries bypass morphology (`*` morphcode

[†] The authors are greatly indebted to Paul Meurer, who kindly shared his implementation of the C library interface for Georgian, which, in the absence of documentation of this feature of XLE, made it possible to develop the current solution for Polish. The authors are also grateful to John Maxwell III for discussion of tokenisation and morphology issues in XLE. The usual disclaimers apply.

[1] In this paper, unless explicitly stated otherwise, Morfeusz refers to Morfeusz 2 – the recent reimplementation of the original Morfeusz [14].

specification in XLE). The information about segmentation and morphosyntactic interpretation of identified segments could be taken from a variety of sources: the previous version of Morfeusz (the predecessor of Morfeusz 2), where segmentation is sometimes ambiguous, while the morphosyntactic interpretation of segments is often highly ambiguous, or from XML files from Składnica treebank [12, 16] or the National Corpus of Polish (NKJP; [8]), which provide unambiguous information about segmentation and morphosyntactic interpretation. The script runs in two modes: interactive, where it runs XLE, intercepts the sentence to be parsed, creates a dedicated lexicon and passes the sentence to XLE for parsing, or in batch mode, where it creates a lexicon for the provided list of sentences – either one file for all sentences or individual files for particular sentences; the obtained lexicon files can be used subsequently with XLE for parsing.

Although such a solution is satisfactory for the specific and restricted purposes of creating subsequent versions of the Polish parsebank when using the disambiguated information from Składnica or NKJP, it is suboptimal for parsing running, unprocessed text: it is incapable of handling ambiguous segmentation (it uses heuristics to choose one segmentation) and, while it can be used with XLE (as described above), it cannot be used to make the grammar available via XLE-Web – an INESS ([10]; `http://iness.uib.no/`) web-service for parsing using XLE: the Python script for creating the lexicon on the fly cannot be used in XLE-Web (without introducing modifications in INESS) and a lexicon containing all Polish forms is too big to load (not to mention doing so in reasonable time limits).[2]

It was therefore decided to devise a solution following the general architecture assumed in XLE, which requires specifying transducers that will handle tokenisation and morphological analysis of an input sentence. It is a common practice in LFG grammars developed within the XLE framework to build such a transducer using the XFST tool [1]. Since a high-quality, effective tool that is well-adjusted to Polish is available, such a solution would require a lot of redundant work whose outcome is not guaranteed to be of comparable quality. Instead, an alternative solution was chosen: to use a programming interface provided within XLE that makes it possible to implement a wrapper library in C/C++ that passes the output of an external morphological tool on to the grammar.

2 Interfacing Morfeusz

The basis of the morphological component for POLFIE is Morfeusz, a state-of-the-art morphological analyser for Polish. Morfeusz is built on the grammatical description and linguistic data of *Grammatical Dictionary of Polish* (SGJP; [11]). Its default inflectional dictionary (mapping between word forms and morphological interpretations, i.e. ⟨lemma, tag⟩ pairs), derived from SGJP, contains over 4,000,000 word forms belonging to over 250,000 lemmata. Another crucial com-

[2] In preliminary experiments, soon abandoned due to excessive size of the resulting files, a full-form lexicon for 8 740 most frequent lemmata in a corpus was a 13.4 GB file.

ponent of Morfeusz is its set of hand-crafted segmentation rules, which allow, for instance, to account for situations where some elements (mostly clitics) are treated as separate segments even though they are not separated by whitespace characters.

Segmentation rules also increase the coverage of the analyser beyond dictionary-defined words by providing a limited derivational component. As an example, the words *europoseł* 'member of the European Parliament' or *hiperaktywny* 'hyperactive' are not explicitly accounted for in the inflectional dictionary of Morfeusz. Instead, the dictionary contains prefixes EURO- and HIPER-, and the segmentation rules admit composing those prefixes with any noun or adjective. In conjunction with the presence of noun POSEŁ 'MP' and adjective AKTYWNY 'active' in the dictionary, the whole mechanism makes it possible to correctly analyse *europoseł* and *hiperaktywny*.

Apart from its vast coverage and reliable linguistic data, Morfeusz introduced another very advantageous feature. In its previous version, the analyser was strictly bound to one dictionary and the segmentation rules were hard-coded into its implementation. The reimplementation, however, has a more flexible architecture, making it possible to provide one's own dictionary and/or segmentation rules instead, either obtained by a modification of the default ones, or constructed from scratch. Morfeusz can be used either as a stand-alone program or, more conveniently from a programmer's point of view, its core library can be called directly from C/C++ or Python code. This section presents a morphological component for the Polish LFG grammar that uses all those features of Morfeusz, while keeping in line with the grammar architecture of XLE framework.

SGJP serves as a basis for the tagset of NKJP [7], adopted in turn by POLFIE. Although similar, the tagset diverges from SGJP in several respects. This is where the aforementioned flexibility of Morfeusz proves very useful – it makes it possible to introduce some grammar- and tagset-specific modifications to the original, SGJP-based dictionary. These include some systematic changes (such as the reduction of SGJP's 9 grammatical genders to NKJP's 5 or a different analysis of some numerals) as well as a few word-specific adjustments. The modified version of the dictionary is kept consistent with updates of the original one. The compiled dictionary's size is about 7.6 MB.

The XLE grammar architecture assumes two steps of processing an input sentence before it is analysed using the grammar rules. The first step is tokenisation: a string of characters is divided into tokens representing particular words. Each token output at this stage is subsequently passed on to the morphological analysis step, where it is associated with a word lemma and morphological tags. The form of the token determines the string that appears in the corresponding LFG c-structure leaf, whereas the morphological information is used in the grammar rules to construct an appropriate analysis. Such a division of tasks between the tokeniser and the analyser is quite the opposite of the architecture of Morfeusz. Due to inflectional features and orthographic rules of Polish, morphological interpretation of some segments depends on the segmentation itself – it is therefore natural and convenient to tightly couple the two steps. As a result, Morfeusz processes an input text in a single run, yielding a so-called morphological analysis graph, representing the (possibly ambiguous) segmentation of the text together with the possible morphological interpretations of particular segments. Figure 1 shows an analysis graph produced by Morfeusz for the input text *Czym rzuciła?*

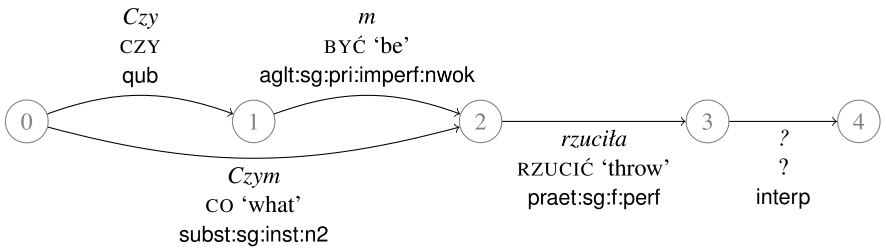


Figure 1: Morfeusz analysis of *Czym rzuciła?*

Depending on the chosen segmentation, the question has two readings:

(1) *Czym* *rzuciła?*
what.SG.INST.N threw.SG.F
'What did she throw?'

(2) *Czy* *m* *rzuciła?*
QPART SG.1 threw.SG.F
'Did I throw (something)?'

The interpretation of the sentence *Czym rzuciła?* depends on whether the word *Czym* is analysed as one or two segments: in (1) it is a form of the interrogative pronoun CO 'what'. By contrast, in (2) the word *Czym* is split into two segments, where the first is a *yes/no* question marker (*Czy*), while the other is an agglutinate form of the verb BYĆ 'be' – it carries information about the person (first) and number (singular) of the subject of the lexical verb (*rzuciła*).

In order to adapt Morfeusz for use with the grammar, two wrapper libraries implementing the interface required by XLE were created. Both libraries make internal calls to Morfeusz, but they process the results differently in order to provide separate tokeniser and analyser functionality as expected by XLE.[3] The tokeniser takes as its input the whole sentence to be parsed, and uses Morfeusz to obtain its segmentation. The analysis graph produced by Morfeusz is translated into a regular expression as specified by XLE's interface, taking into account any ambiguities at the level of segmentation. The tokeniser library also deals with comma haplology as proposed in [4].[4] For example, the regular expression for the segmentation

[3] The current solution does not include any guesser functionality: only words recognised by Morfeusz are given morphosyntactic analyses.

[4] The general idea is to assume, in grammar rules, that all parenthetical clauses are delimited by commas both at the beginning and at the end. In written text this is usually not the case due to orthographic rules, therefore, the tokeniser inserts "optional commas" where they could be expected by the grammar. Following an example for English from [4], the sentence *Find the dog, a poodle.* (compare with *Find the dog, a poodle, now!* where both delimiting commas are present in written text) can thus obtain an alternative tokenisation ‹Find› ‹the› ‹dog› ‹,› ‹a› ‹poodle› ‹,› ‹.›, allowing the parser to treat *a poodle* as a comma-delimited clause. For simplicity, the inserted commas are not shown in the discussed example.

provided by the analysis from Figure 1 would be:

(3) `(‹Czy› ‹+m›|‹Czym›) ‹rzuciła› ‹?›`

In this example, the segmentation ambiguity is reflected by the alternative `(‹Czy› ‹+m›|‹Czym›)`, and the whole expression encodes the two possible segmentations of the sentence *Czym rzuciła?*:

(4) `‹Czy› ‹+m› ‹rzuciła› ‹?›` (5) `‹Czym› ‹rzuciła› ‹?›`

The tokens output by the tokeniser are then, separately, passed by XLE to the analyser library. The analyser, in turn, once again runs Morfeusz, this time on individual tokens. Such architecture introduces an artificial division of the actions normally carried out simultaneously by the Polish morphological analyser. One negative consequence of such a solution is that when Morfeusz is called from the analyser library, it only has access to a single segment, without the context provided by surrounding text that would normally be available to segmentation rules implemented in Morfeusz. In order to prevent information loss between the two stages, some auxiliary modifications were introduced at the tokenisation and analysis level.[5] The result of analysing the individual tokens from the example *Czym rzuciła?* would be the following morphology outputs:

(6)
```
czy +qub                       (‹Czy›)
być +aglt:sg:pri:imperf:nwok   (‹+m›)
co +subst:sg:inst:n2           (‹Czym›)
rzucić +praet:sg:f:perf        (‹rzuciła›)
?  +interp                     (‹?›)
```

In order to illustrate the benefit from incorporating the Polish-specific segmentation mechanism of Morfeusz into the grammar, the LFG analyses for the input text `Czym rzuciła?` obtained using two different variants of POLFIE are discussed below.[6] The two variants produce the same number of analyses, identical f-structures and identical c-structuresas far non-terminal nodes are concerned. However, there is an important difference at the level of c-structure terminals.

The first variant uses the wrapper analyser library with a tokeniser originally implemented in XFST by Ron Kaplan for the English LFG grammar (see [4]). Since this tokeniser cannot divide a word like *Czym* into two tokens (the only tokenisation being `‹Czym› ‹rzuciła› ‹?›`), additional segmentation is performed at the morphological analysis stage. For any token passed to the analyser library that can be interpreted as more than one segment, the ambiguous segmentation is retrieved from Morfeusz and reflected in the morphology outputs:

[5] One such modification is adding '+' to *m* in the discussed tokenisation example – this solution retains the information that the segment is a clitic and therefore makes it possible to block its "standalone" interpretation (METR 'metre').

[6] INESS' XLE-Web component was used for visualising and disambiguating the structures.

(7)
```
czy +qub być +aglt:sg:pri:imperf:nwok|co +subst:sg:inst:n2
rzucić +praet:sg:f:perf
? +interp
```

Note that though, in this way, all the correct lemmata and morphosyntactic tags are obtained, there is no means of associating them with different tokenisations of the input sentence. All three ⟨lemma, tag⟩ pairs generated for the token ‹ `Czym` › (first row of (7)) are mapped by XLE to that same token. The result is that, with the analysis for interpretation (2) chosen, corresponding to the `czy +qub być +aglt:sg:pri:imperf:nwok` option, both preterminal c-structure nodes, QUB[int] and AGLT, are associated with the terminal *Czym* (see Figure 2).

In the second variant, which is the solution offered in this paper, both tokeniser and analyser libraries use Morfeusz. Since the tokeniser has already handled any segmentation ambiguities, the analyser library is configured to treat any token (with an exception explained in Section 3) as "unambiguous": even if the analysis from Morfeusz contains ambiguous segmentation (as in the case of ‹ `Czym` ›), only the morphological interpretations pertaining to the whole token string are returned.[7] In this way, only the ‹ `Czy` › and ‹ `+m` › tokens (not ‹ `Czym` ›) contribute to the `czy +qub` and `być +aglt:sg:pri:imperf:nwok` interpretations respectively. The resulting LFG analysis for interpretation (2) is presented in Figure 3.

Figure 4 shows the analysis for interpretation (1), where *Czym* is one segment. Since this interpretation does not involve any Polish-specific segmentation phenomena, both variants of the grammar yield identical structures.

3 Multi-word expressions

The current version of POLFIE supports a small number of MWEs. The plans for further development of the grammar include enlargement of this stub MWE component using information gathered from resources such as Walenty or SEJF (an MWE dictionary, [3]). It is not obvious that all Polish MWEs should be analysed at the tokenisation and morphology levels, as is often assumed in LFG grammar architectures. Many MWEs are clearly compositional syntactically, thought not semantically. One might therefore want to analyse their syntactic structure, and use the resources mentioned above to mark them as semantically non-compositional for the purposes of future semantic processing. It seems, nevertheless, uncontroversial that some types of MWEs, mostly the fixed, non-inflecting sequences, can and should be handled at the stage of tokenisation and morphological analysis. The currently supported MWEs, mostly belonging to closed grammatical classes (such as conjunctions, complementisers and prepositions), have such characteristics – it was therefore decided to analyse them in the relevant libraries.

[7] The interpretations themselves can nevertheless be ambiguous and they mostly are due to homonymy and ubiquitous syncretism in Polish. For example, the output for the token ‹ `pośle` › would be `posłać +fin:sg:ter:perf|poseł +subst:sg:loc:m1|poseł +subst:sg:voc:m1` since the word *pośle* could be a form of either POSŁAĆ 'to send' or POSEŁ 'MP'.

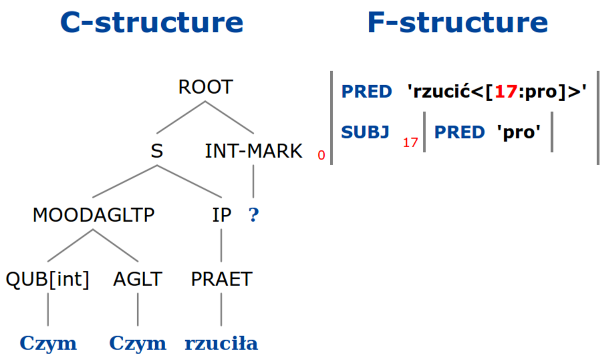


Figure 2: *Czym rzuciła?* – structures for interpretation (2), Ron Kaplan's tokeniser.

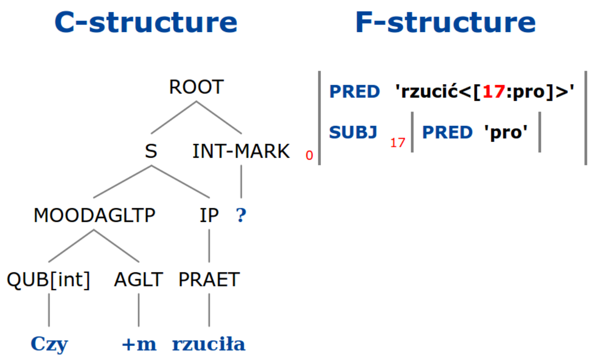


Figure 3: *Czym rzuciła?* – structures for interpretation (2), Morfeusz-based tokeniser.

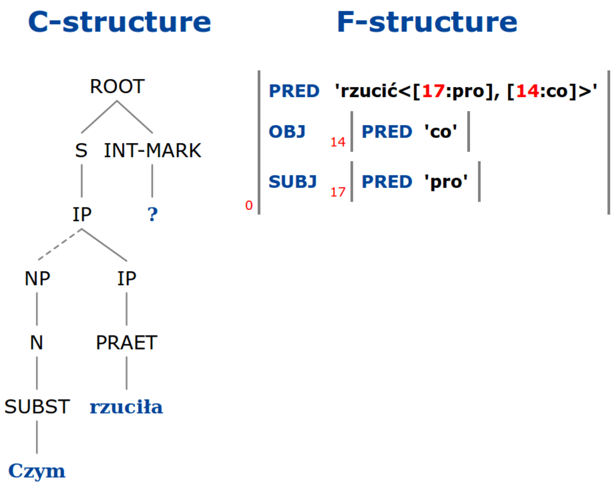


Figure 4: *Czym rzuciła?* – structures for interpretation (1).

Since Morfeusz, by design, does not admit segments longer than those delimited by whitespace characters, it is not possible to analyse MWEs by specifying them in Morfeusz dictionary. A small dictionary of multi-word expressions that is used by the tokeniser and analyser libraries was created for this purpose. For each MWE to be analysed, there is a specification of the lemmata and morphosyntactic tags of its component words as well as the lemma and morphosyntactic tag of the whole expression.[8] Two examples of entries in the dictionary are:

(8) `w +prep:loc:nwok czas +subst:sg:loc:m3 : 'w czasie' +prep:gen`

(9) `a +conj nie +qub : 'a nie' +conj`

The entry in (8) corresponds to W CZASIE ‘during, lit. in time (of)’, which is treated in POLFIE as a preposition selecting for the genitive case. The left-hand side of this entry specifies a sequence of two tokens having `w +prep:loc:nwok` and `czas +subst:sg:loc:m3` respectively among their Morfeusz analyses. Whenever such a sequence is encountered, it obtains an additional analysis as one token,[9] with a morphological interpretation `'w czasie' +prep:gen`. The entry in (9) corresponds to the conjunction A NIE ‘but/and not’.

The general architecture of tokenisation and morphology libraries in XLE and the way how Morfeusz handles whitespaces makes introducing some redundancy into the analysis of MWEs inevitable. First, the tokeniser library tries to match the consecutive segments returned by Morfeusz with the MWE entries. Whenever a match is found, the segments obtain an alternative tokenisation as a single token. For example, the sentence (10) would obtain the tokenisation in (11):

(10) *W czasie podróży dużo czytał.*
in.LOC time.LOC travel.GEN much read
‘During the travel, he read a lot.’

(11) `(‹W› ‹czasie›|‹W czasie›) ‹podróży› ‹dużo› ‹czytał› ‹.›`

Second, if the analyser library is given a token containg a space, it tries once again to match its analysis with the sequences of interpretations specified in the MWE dictionary, this time using the second part of a matched entry to provide a suitable lemma and morphosyntactic tag.

4 Abbreviations

Another type of token that is handled in a special way are abbreviations. Morfeusz makes a distinction between punctuated abbreviations (assigned a `brev:pun` tag and required to be followed by a period) and non-punctuated ones (tagged as

[8] The reason for operating on lemmata and tags returned by Morfeusz rather than simply specifying the MWE’s text form is that the former solution makes it possible to benefit from the analyser’s robust handling of lowercase/uppercase orthographic variations.

[9] The analysis as separate tokens is also kept as it could also be valid.

`brev:npun`). Both types of abbreviations are lemmatised to the lemma of their corresponding full form. For instance, the word *ul* can be interpreted either as a form of UL 'beehive' or, when followed by a period, as an abbreviated form of ULICA 'street'. Polish orthographic rules generally require that abbreviations must be punctuated if their last letter is different from the last letter of the word form they stand for, it therefore follows that a non-punctuated occurrence of *ul* should not be interpreted as abbreviated form of ULICA – it can only be a form of UL. The segmentation rules of Morfeusz take this into account: its analyses of *ul* and *ul.* are shown in Figure 5. Another example of a punctuated abbreviation is *prof.* 'professor'. Since, unlike *ul*, the word *prof* has no other meanings in Polish, its non-punctuated occurrences are marked by Morfeusz as unrecognised by assigning them an `ign` (ignotum) tag. Analyses of *prof* and *prof.* are shown in Figure 6. The non-punctuated abbreviations, such as *wg* (WEDŁUG 'according to') are interpreted as `brev:npun` in any context.

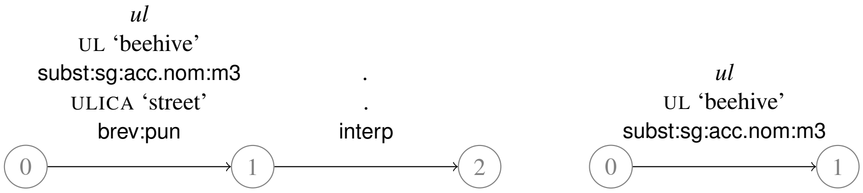


Figure 5: Morfeusz analyses of *ul.* (left) and *ul* (right).

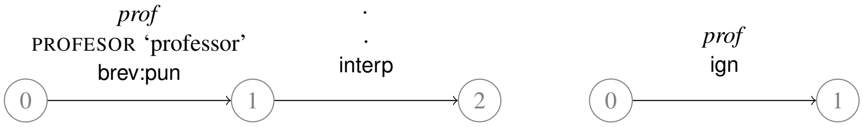


Figure 6: Morfeusz analyses of *prof.* (left) and *prof* (right).

The way Morfeusz analyses abbreviations is incompatible with POLFIE grammar in two respects. First, the grammar requires abbreviations to be assigned not only the lemma, but also morphosyntactic tags of their full forms. For instance, the expected analysis of *wg* is `według +prep:gen`, not `według +brev:npun`. Second, the grammar rules assume that a comma following a punctuated abbreviation is a part of its token, not a separate one. For instance, the expected tokenisation of *ul* is ‹`ul.`›, not ‹`ul`› ‹`.`›.[10]

The first issue is addressed by modifying the default dictionary of Morfeusz: the `brev:pun`/`brev:npun` entries are replaced with full morphosyntactic tags of the corresponding forms. For this purpose, a mapping analogous to the one defined in the grammar of *Świgra*, a DCG parser of Polish [13], is used.

[10] Such a way of analysing abbreviations is implemented in Ron Kaplan's tokeniser that was used in the previous version of POLFIE.

The second issue requires more complicated modifications, covering also the problem of period haplology (a period following a punctuated abbreviation can at the same time be a sentence-ending period, see [4]). The proposed solution works as follows: the first step is to recognise fragments of the analysis graph returned by Morfeusz that correspond to punctuated abbreviations and their following periods. In case of a segment that only has an abbreviation interpretation, like *prof*, the period is appended to its token, and an additional, optional period is added (marked with `?`; it would only be consumed by the grammar rules when occurring at the end of a sentence). For instance, the tokenisation for *prof.* would be ‹`prof.`› ‹`.`›`?`. For segments that also have any non-abbreviation interpretations, the original segmentation from Morfeusz is kept as an alternative tokenisation variant. As an example, the tokenisation for *ul.* would be ‹`ul.`› ‹`.`›`?|`‹`ul`› ‹`.`›.

Second, an auxiliary Morfeusz dictionary is introduced – it is dedicated specifically to punctuated abbreviations. In this dictionary, the entries for those abbreviations contain periods at their ends, and therefore the corresponding tokens passed from the tokeniser library can be analysed as required by the grammar. The reason for introducing this additional dictionary is to avoid interference with the original segmentation mechanisms of Morfeusz that are preserved in the "main" dictionary and make it possible to properly recognise abbreviation-punctuating periods in the tokeniser library. The main and auxiliary dictionaries are combined in the `ANALYZE USEFIRST` section of the morphology configuration.

5 Adapting the grammar

To use morphology in XLE, one must create sublexical rules – the right-hand side contains the stem (`Vsub_BASE`; verbal stem), which introduces constraints appropriate for the given lemma (this includes valence information – arity of the predicate and associated constraints, if any – as well as lexicalised constraints, if applicable), and tags returned by the analyser (`Vpraet_SFX_BASE`; l-participle tag), which introduce morphosyntactic information associated with the given form; the left-hand side corresponds to the resulting category (`PRAET`; l-participle):

(12) `PRAET --> Vsub_BASE   Vpraet_SFX_BASE.`

To use such a rule, lexical entries must be created for stems and tags used in sublexical rules. The example provided in (13) is the lexical entry for RZUCIĆ 'throw' – the first field corresponds to the lemma (`rzucić`), the second one is the category (`Vsub`, verbal), the third is the morphcode signalling that it passes through morphology (`XLE`, unlike `*`) and the last one contains constraints imposed by the particular entry – in this case it is a call to the template `@(rzucić-Walenty)`, which contains valence schemata appropriate for this particular verb:

(13) `rzucić   Vsub   XLE   @(rzucić-Walenty).`

In POLFIE, valence templates are zero-argument templates – this is why the template call such as `@(rzucić-Walenty)` consists of the template name exclusively.

Each valence template definition rewrites to a disjunction of converted valence schemata from Walenty for the given lemma, where each such disjunct consist of two parts: the specification of the `PRED` attribute, which lists arguments of the relevant predicate (provided below), and a set of constraints related to these arguments (not shown below due to space limits – `[constraints]` is a placeholder).

(14)
```
rzucić-Walenty =
  { (^ PRED)='rzucić<(^ SUBJ)(^ OBJ-TH)>' [constraints]
  | (^ PRED)='rzucić<(^ SUBJ)(^ COMP)>' [constraints]
  | ... }.
```

Next, lexical entries are created for tags returned by the analyser:

(15)
```
+praet:sg:f:perf   Vpraet_SFX   XLE   @(PRAET sg f perf).
```

The structure of lexical entries of tags is the same as discussed above: the first field in (15) is the full morphosyntactic tag (`+praet:sg:f:perf`), the second is the category (`Vpraet_SFX`), the third is the morphcode marking that it passes through morphology (`XLE`) and the last one introduces constraints (`@(PRAET sg f perf)` is a template call, see the following discussion).

The approach to tags presented here seems to be different from mainstream one in that the analyser returns a single morphosyntactic tag for each interpretation of a particular segment – these tags are created in accordance with the NKJP tagset [7], a positional tagset where the first element of the tag (tag parts are separated by the `:` symbol) is the part of speech (`praet`, l-participle, in (15)) and then values of morphosyntactic categories appropriate for it (if any) follow: in (15) these include `sg` for singular number, `f` for feminine gender and `perf` for perfective aspect.

Using such "glued" tags makes it possible to rewrite them directly to calls to part of speech templates: the entry of `+praet:sg:f:perf` contains the call `@(PRAET sg f perf)`, where particular parameters of the `PRAET` template set appropriate values of relevant attributes (as discussed above). This makes it possible to avoid creating separate lexical entries for values of the same attribute used with different parts of speech – for instance, both verbs and nouns are specified for number, but under the mainstream LFG analysis number in nouns sets the number value of the noun, but in verbs it introduces a constraint on the number of the verb's subject (rather than the verb itself). Under the current solution such differences are accounted for inside the definitions of particular part of speech templates.

The last issue is creating the lexical entries for lemmata – as mentioned above, lexical entries introduce valence constraints and possibly lexicalised constraints. The current grammar makes extensive use of the `-unknown` lexical entry, which serves the purpose of creating lexical entries for lemmata not listed in the lexicon because their behaviour is fully regular – as opposed to items introducing lexicalised constraints, which must be listed explicitly in the lexicon. Below is a fragment of the `-unknown` entry handling adjectives (`Asub`) and adverbs (`ADVsub`):

(16)
```
-unknown   Asub     XLE   @(ZERO-OR-PRED %stem);
           ADVsub   XLE   @(ZERO %stem).
```

The first subentry (`Asub`) introduces zero-argument (attributive) or one-argument SUBJ (predicative) subcategorisation for adjectives; the second one (`ADVsub`) assigns zero-argument valence specification for adverbs. Since the subentries of `-unknown` are matched against the category of the segment, which is in turn determined by the interpretation from the analyser (returned tags), there is no risk of assigning adverbial subcategorisation to a segment which is not an adverb.

As a result, only "special" lemmata are listed in the lexicon – these include, for instance, *n*-words, which introduce constraints looking for sentential negation in the relevant domain (in Polish, *n*-words need to be licensed).

6 Conclusion

This paper offered a method of integrating the Polish LFG grammar with a state-of-the-art morphological analyser Morfeusz. It improves the grammar and the parsebank by making it possible to fully use the Polish-specific mechanisms, such as ambiguous segmentation, implemented in Morfeusz. At the same time, it keeps the grammar compliant with XLE architecture and compatible with XLE-based tools.

Acknowledgements

This work is partially financed by the Polish Ministry of Science and Higher Education within the CLARIN ERIC programme 2015–2016 (`http://clarin.eu`), and partially by the European Union from resources of the European Social Fund *Project PO KL "Information technologies: Research and their interdisciplinary applications"*.

References

[1] Kenneth R. Beesley and Lauri Karttunen. *Finite State Morphology*. CSLI Publications, Nancy, France, 2003.

[2] Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors. *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, Reykjavík, Iceland, 2014. ELRA.

[3] Monika Czerepowicka and Agata Savary. SEJF – a grammatical lexicon of Polish multi-word expressions. In Zygmunt Vetulani and Joseph Mariani, editors, *Proceedings of the 7th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 224–228, Poznań, Poland, 2015.

[4] Ron Kaplan, John T. Maxwell, Tracy Holloway King, and Richard Crouch. Integrating finite-state technology with deep LFG grammars. In *Proceedings of the Workshop on Combining Shallow and Deep Processing for NLP (ESSLLI)*, 2004.

[5] Agnieszka Patejuk and Adam Przepiórkowski. Towards an LFG parser for Polish: An exercise in parasitic grammar development. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012*, pages 3849–3852, Istanbul, Turkey, 2012. ELRA.

[6] Agnieszka Patejuk and Adam Przepiórkowski. Synergistic development of grammatical resources: a valence dictionary, an LFG grammar, and an LFG structure bank for Polish. In Verena Henrich, Erhard Hinrichs, Daniël de Kok, Petya Osenova, and Adam Przepiórkowski, editors, *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT 13)*, pages 113–126, Tübingen, Germany, 2014. Department of Linguistics (SfS), University of Tübingen.

[7] Adam Przepiórkowski. A comparison of two morphosyntactic tagsets of Polish. In Violetta Koseska-Toszewa, Ludmila Dimitrova, and Roman Roszko, editors, *Representing Semantics in Digital Lexicography: Proceedings of MONDILEX Fourth Open Workshop*, pages 138–144, Warsaw, 2009.

[8] Adam Przepiórkowski, Mirosław Bańko, Rafał L. Górski, and Barbara Lewandowska-Tomaszczyk, editors. *Narodowy Korpus Języka Polskiego [Eng.: National Corpus of Polish]*. Wydawnictwo Naukowe PWN, Warsaw, 2012.

[9] Adam Przepiórkowski, Elżbieta Hajnicz, Agnieszka Patejuk, Marcin Woliński, Filip Skwarski, and Marek Świdziński. Walenty: Towards a comprehensive valence dictionary of Polish. In Calzolari et al. [2], pages 2785–2792.

[10] Victoria Rosén, Koenraad De Smedt, Paul Meurer, and Helge Dyvik. An open infrastructure for advanced treebanking. In Jan Hajič, Koenraad De Smedt, Marko Tadić, and António Branco, editors, *META-RESEARCH Workshop on Advanced Treebanking at LREC2012*, pages 22–29, Istanbul, Turkey, 2012.

[11] Zygmunt Saloni, Marcin Woliński, Robert Wołosz, Włodzimierz Gruszczyński, and Danuta Skowrońska. *Słownik gramatyczny języka polskiego*. Warsaw, 2nd edition, 2012.

[12] Marek Świdziński and Marcin Woliński. Towards a bank of constituent parse trees for Polish. In *Text, Speech and Dialogue: 13th International Conference, TSD 2010, Brno, Czech Republic*, Lecture Notes in Artificial Intelligence, pages 197–204, Berlin, 2010. Springer-Verlag.

[13] Marcin Woliński. *Komputerowa weryfikacja gramatyki Świdzińskiego*. Ph.D. dissertation, Institute of Computer Science, Polish Academy of Sciences, Warsaw, 2004.

[14] Marcin Woliński. Morfeusz — a Practical Tool for the Morphological Analysis of Polish. In Mieczysław Kłopotek, Sławomir T. Wierzchoń, and Krzysztof Trojanowski, editors, *Intelligent information processing and web mining*, pages 503–512. Springer-Verlag, 2006.

[15] Marcin Woliński. Morfeusz reloaded. In Calzolari et al. [2], pages 1106–1111.

[16] Marcin Woliński, Katarzyna Głowińska, and Marek Świdziński. A preliminary version of Składnica—a treebank of Polish. In Zygmunt Vetulani, editor, *Proceedings of the 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 299–303, Poznań, Poland, 2011.

Semantic Role Annotation in BulTreeBank

Petya Osenova and Kiril Simov

Linguistic Modelling Department
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
E-mail: {petya|kivs}@bultreebank.org

Abstract

In this paper we present an approach for semi-automatic annotation of semantic roles over a syntactically and sense annotated corpus – BulTreeBank. The annotation is facilitated by the available annotation of syntactic structure, the semantic class of the verb, senses of the argument elements of the verb and valencies. We present the annotation procedure, some preliminary results and report on the problems encountered in the process.

1 Introduction

In the era of cross-resourced data, such as BabelNet[1], UBY[2] and Predicate Matrix[3], Linguistic Linked Open Data[4], among others, the treebanks (as well as parsebanks) are becoming valuable containers of rich language knowledge, which adds to the syntactic structure various types of semantics and discourse information. One piece of such semantic information are the semantic roles. In spite of their controversial and non-homogeneous nature, semantic roles remain the most common interface between the grammatical functions and the semantics of the sentence.

The most popular approaches to semantic roles have been implemented in initiatives like VerbNet[5] and FrameNet[6].

In this paper we present our methodology of semantic role annotation in the BulTreeBank original format. The focus is on the argument roles only. The adjunct roles will be handled later. The step of semantic role annotation comes after some preliminary work, which included: the creation of valency dictionary (reported in [5]) and annotation with WordNet senses (reported in [6]).

[1] http://babelnet.org/

[2] https://www.ukp.tu-darmstadt.de/data/lexical-resources/uby/

[3] http://adimen.si.ehu.es/web/PredicateMatrix

[4] http://linguistic-lod.org/llod-cloud

[5] http://verbs.colorado.edu/ mpalmer/projects/verbnet.html

[6] https://framenet.icsi.berkeley.edu/fndrupal/

The paper is structured as follows: in section 2 the related works are discussed; in section 3 the prior annotations are described; section 4 presents our strategy for semantic role annotation; section 5 reports on the preliminary results; section 6 concludes the paper.

2 Related Work

One of the most notable treebanks, annotated with semantic roles, is PropBank. PropBank[7] results in adding a layer of predicate-argument relations (semantic roles) to the syntactic trees of the Penn Treebank. The syntactic structure of the trees in Penn Treebank is used in the PropBank annotation to assign semantic labels to nodes in the trees. The Penn Treebank does not distinguish the different semantic roles played by a verb's grammatical functions. Since the same verb used with the same syntactic subcategorisation, but with different semantic role sets can assign different semantic roles, roles cannot be deterministically added to the Treebank by an automatic process. Thus every instance of every verb in the treebank is covered.

The goal behind PropBank project was the creation of a broad-coverage hand annotated corpus of semantic roles together with the verb alternations. The annotation relies on the linking between semantic roles and syntactic realization. The syntactic frames are a direct reflection of the underlying semantics [4]. PropBank uses a traditional set of semantic roles (such as, Agent, Patient, Theme, etc.). An individual verb's semantic arguments are numbered, beginning with 0. For a particular verb, Arg0 is generally the argument exhibiting features of a prototypical agent while Arg1 is a prototypical patient or theme. However, it uses a rich set of adjunct roles (for example, PropBank's ArgM Modifier Roles include: LOC: location; EXT: extent; DIS: discourse connectives; ADV: general-purpose; NEG: negation marker; MOD: modal verb; CAU: cause; TMP: time; PCN: purpose; MNR: manner; DIR: direction). Other treebanks follow this model, too. For example, EPEC-RolSem [2] is a Basque corpus labeled at predicate level following the PropBank-VerbNet model, but being adapted to the specificities of the language.

In Prague Dependency Treebank (PDT) the semantic roles are presented at the tectogrammatical level. In contrast to PropBank, here all the semantic roles are named (the argument-related ones as well as the adjunct-related ones) – Actor, Patient, Addressee, etc.

When comparing the approaches in PropBank and PDT, the local semantics of verbs in PropBank contrasts with the global semantics of verbs in PDT, because the global semantics reflects a specific grammar framework. For example, the relations in PDT are richer than these in PropBank, which causes difficulties in getting optimal compatibility of data annotation models. PDT has implemented its theory from the start, while PropBank is a build-on over Penn Treebank. PropBank relies on bigger generalizations, using notations, such as Arg0, Arg1, Arg2, etc., while PDT

[7] http://verbs.colorado.edu/ mpalmer/projects/ace.html

introduces a rich set of roles. Despite different levels of underspecification, both resources make use of rich valence and beyond-valence lexicons (PropBank – from VerbNet and FrameNet; PDT – from their in-house constructed lexicon Vallex[8]). It should be noted that nowadays these two types of treebanks have become best practices for creating similar resources for other languages.

Comparing our approach to the ones described above, the following facts can be observed: similarly to the PropBank strategy, we added a predicate-argument layer over a syntactically annotated corpus (BulTreeBank); similarly to the PDT strategy, the added semantic roles have been labeled. In contrast to both strategies we consider only core arguments, such as the subject and the direct/indirect objects, but not adjuncts. Also, similarly to the PropBank approach, we use the VerbNet semantic roleset. However, it is used through the mappings of the Bulgarian verbs to the Princeton WordNet semantic classes with the respective adjustments.

The idea of using verb hierarchies to assign semantic interpretations is not new. For example, [1] exploits the WordNet hierarchy of nouns and their syntactic relations to assign thematic roles to the predicate's arguments. [3] also uses the WordNet hierarchy for a semi-automatic classification of verbs with respect to Levin's verb classes. Although these works are different from our task, we gain insights from them with respect to the principles of inheritance and the combination of syntactic with semantic knowledge.

3 Prior Annotations

The creation of valency dictionary and the WordNet-based sense annotation approach were discussed in our previous works (see above). Here we comment in brief the main points that are related to the process of semantic role annotation.

The valency frames were extracted from the treebank, manually processed with verb senses and detailed participants with respect to the usage, and then returned back into the treebank.

The sense annotation was performed in the following stages: (i) Mapping the definitions of a Bulgarian explanatory dictionary to the intersected senses of Core and Base Concepts in Princeton WordNet, where mappings were manually checked and curated according to the selection of the correct sense, addition of a sense or update of a definition; (ii) Mapping nouns, verbs, adjectives and adverbs from the treebank to WordNet.

In table 1 some statistics is given on the number of mapped to WordNet parts-of-speech.

Needless to say, verbs are the most important part-of-speech for the task of semantic role annotation. When mapped to the WordNet senses, they receive also a semantic class, which, together with the valency frames, helps in the selection of the appropriate semantic roles.

[8] http://ufal.mff.cuni.cz/vallex/2.5/doc/home.html

	Adj	**Adv**	**Noun**	**Verb**	**Total**
Number of Tokens	17 304	10 728	37 330	14 341	79 703

Table 1: Statistics over the mapped parts-of-speech.

4 Semantic Roles: Approach and Procedure

Our approach relies on the following prerequisites: (i) the availability of the syntactic functions in the parsed data (Subject, Direct Object and Indirect Object); (ii) the restrictions of the valency frames and the senses, as well as (iii) the semantic classes of the verbs.

The syntactic functions are provided by the treebank. The valency frames come from the valency dictionary of Bulgarian [5]. At the moment it contains 4113 valency frames coupled with the respective meaning and it covers 1903 lemmas. The semantic classes of the verbs have been transferred by the mappings of the Bulgarian valency lexicon to the Princeton WordNet.

The statistics of the verb lemmas with respect to the verb semantic classes in WordNet (WN-VC) is given in Table 2.

Verb Class	**Number**	**Verb Class**	**Number**
Verb.Communication	283	Verb.Creation	95
Verb.Social	222	Verb.Perception	86
Verb.Stative	219	Verb.Competition	63
Verb.Motion	204	Verb.Emotion	53
Verb.Cognition	203	Verb.Body	41
Verb.Change	184	Verb.Weather	14
Verb.Possession	130	Verb.Consumption	13
Verb.Contact	97		
		Total	1907

Table 2: Statistics over the WordNet verb classes in the treebank.

As it can be seen, the most frequent class is verb.communication. The next most frequent ones are verb.social and verb.stative. This situation is due to the fact that the treebank consists of predominantly newsmedia texts plus also some fiction and administrative documents.

Our semantic role set follows strictly the stipulations in VerbNet[9]. When information is missing, other related resources are consulted, such as FrameNet[10]. First, a very general frame is assigned to a verb class. For example, the verbs of consumption and change have as typical participants AGENT and PATIENT;

[9] http://verbs.colorado.edu/verb-index/

[10] https://framenet.icsi.berkeley.edu/fndrupal/index.php?q=frameIndex

the verbs of perception and cognition have as typical participants EXPERIENCER and THEME, etc. Then, depending on the specific valency frame, some more fine-grained or other roles can be added. Let us consider the weather verbs. Most of them are intransitive and assign their subject the THEME role (thunder, glow, shine, erupt etc.). However, some of them can be transitive as well, assigning the role of CAUSE (kindle, set-on-fire) to their subject and the role of THEME to their direct object. Now let us also consider the body verbs. In the usage with more grammatical roles their subjects can take two roles – AGENT (wear, dress) or EXPERIENCER (sunburn, feel). The direct objects take THEME (wear, take-off, infect) or PATIENT (injure, run-over). The indirect objects can take the roles of COMMUNICATION (laugh at), STIMULUS (sicken), MANNER (feel, act).

The procedure of annotation is as follows:

1. **Selection of WordNet Verb Class.** Our assumption is that all verbs in each of these general classes share common participants in the corresponding events.

2. **Creation of a Hierarchy for Valency Frames.** For the selected WN-VC all the verb synsets in the class are collected. Then, for these synsets all valency frames are identified. They are organized in a hierarchy according to the following heuristics: Subject precedes Direct Object, and Direct Object precedes Indirect Object. According to these heuristics a valency frame with Subject and Direct Object elements is considered as more general to a frame with Subject, Direct Object and Indirect Object elements. Additionally, semantic restrictions are applied to the elements as an additional filter. For a given WN-VC the valency frame hierarchy could have more than one general frame.

3. **Initial Semantic Role Assignment.** Each semantic role is assigned starting from the top frame for each valency hierarchy. We follow the above mentioned principle for typical participants.

4. **Sentence Annotation.** The valency lexicon and the BTB WordNet have been constructed on the basis of the examples taken from the treebank. In this way, for each verb occurrence in the treebank we know the sense and the frame appropriate for its usage. Through the assigned semantic roles to the corresponding valency frame we transfer the semantic roles to the corresponding constituents in the tree of the verb occurrence.

5. **Inspection of the Annotated Sentences.** Each assigned role in the treebank has been manually checked. If necessary, the assigned role might be changed. In such cases the changes are registered in the sentence as well as in the hierarchy of valency frames.

Let us take the verb пия (drink). It is mapped to the class verb.consumption via the Princeton WordNet. In its valency frame SUBJ drinks DObj it has subject-agent and direct object-patient: [Subj=Agent; DObj=Patient]. This case is a straightforward one, because the semantic roles mostly coincide with the prototypical ones. The only cases to be considered further are the passive sentences, since BulTreeBank does not have special annotations over the passive constructions and also the valency lexicon provides the set of semantic roles over the active voice predicates. When we go down the hierarchy we find the verb изпиша in the meaning of 'write out some space'. For example, Той изписа листа (He has written the sheet of paper out.) In this case the semantic frame has been changed to [Subj=Agent; DObj=Asset]. This change is then inherited down the hierarchy.

Another example is the verb пламна, which has two meanings – flare and blush. Thus, it has been mapped to two classes – verb.weather and verb.body. These mappings separate the frames for the two meanings in different hierarchies. In the class verb.weather for intransitive verbs the subject role is Theme. Thus, this role is assigned to the verb: [Subj=Theme]. Going down in the hierarchy for transitive verbs the frame is changed to [Subj=Cause; DObj=Theme], where the role Theme is assigned to the direct object and this changes the role for the subject to Cause. When this change is done, it is applied to all transitive verbs of this class. In the class verb.body the prototypical semantic frame is Subj=Agent and DObj=Patient. But for this verb a change is necessary to the prototypical frame because its subject can be further specified as experiencer and its indirect object as stimulus. After this modification all verbs from the hierarchy of the valency frame (SUBJ blushes INOBJ) will receive the semantic frame: [Subj=Exper; IObj=Stimulus].

5 Preliminary Results

The current status of the semantic roles in the treebank per verb category is presented in Table 3.

All the semantic classes were assigned general frames, which reflect the prototypical participants. For example, the verbs of cognition and perception usually have subjects-experiencers and direct objects-themes; the verbs of change usually have subjects-agents and direct objects-patients, etc.

The general frames were assigned to all the verbs in the treebank, awaiting for further elaboration.

We aimed at assigning semantic roles to the core participants, which are expressed with the following grammatical functions: subject, direct object, indirect object. Further, in the refinement stage, the initially assigned roles are checked and either made more detailed or changed. The adjuncts are not part of this undertaking.

All the prototypical semantic roles have been assigned to the verbs in the treebank, as mentioned above. The general impression is that the main points for changes are as follows: a) passivization (in the text via rules); b) inchoative verbs (in the lexicon); c) addition of a missing core argument (syncronization with the

Category	Frame	Number
verb.body	[Subj=Agent; DObj=Patient] [Subj=Agent] [Subj=Agent; IObj=Communication] [Subj=Agent; IObj=Source] [Subj=Exper; IObj=Stimulus] [Subj=Agent; DObj=Manner] [Subj=Agent; DObj=Theme] [Subj=Exper; DObj=Manner] [Subj=Recipient; DObj=Theme] [Subj=Source; DObj=Theme] [Subj=Agent; DObj=Patient; IObj=Theme] [Subj=Exper; IObj=Theme]	41
verb.change	[Subj=Agent; DObj=Patient; IObj=Goal] [Subj=Theme]	184
verb.cognition	[Subj=Exper; DObj=Theme; IObj=Goal]	203
verb.communication	[Subj=Agent; DObj=Theme]	283
verb.competition	[Subj=Agent; DObj=Patient]	63
verb.consumption	[Subj=Agent; DObj=Patient] [Subj=Asset; IObj=Goal] [Subj=Theme; IObj=Predicate] [Subj=Agent; DObj=Asset] [Subj=Pivot; DObj=Theme]	13
verb.contact	[Subj=Agent; DObj=Patient]	97
verb.creation	[Subj=Agent; DObj=Theme]	95
verb.emotion	[Subj=Exper; DObj=Theme]	53
verb.motion	[Subj=Agent; DObj=Theme]	204
verb.perception	[Subj=Exper; DObj=Theme]	86
verb.possession	[Subj=Locative; DObj=Theme]	130
verb.social	[Subj=Agent; DObj=Patient]	222
verb.stative	[Subj=Agent; DObj=Patient] [Subj=Locative] [Subj=Patient]	219
verb.weather	[Subj=Theme] [Subj=Cause; DObj=Theme]	14

Table 3: Current frame assignment by category and number of verbs in treebank.

valency lexicon); d) refinement of the assigned role (in the lexicon).

Concerning the most frequent types, it seems that the groups of verb.communication, verb.social and verb.stative need more human intervention also due to the availability of many metaphoric and metonymic meanings. On the other hand, the groups of verb.motion and verb.cognition remain closer to their prototypical semantic frames.

As an evaluation of the procedure here we present the processing of three verb classes. The assigned initial general frames have been processed further for the verb categories with relatively small number of frames in the valency lexicon: verb.consumption (2); verb.body (7) and verb.weather (2); 11 in total. These semantic classes also have small number of occurrences in the corpus. After the annotation of the sentences and the correction, the number of the differing frames is 19. Three of the initial role assignments were not used, since they were too general.

The number of the assigned roles is 348. From them 212 are the same as assigned in the beginning. The rest were changed. Thus, in 60.9 % of the cases we did not change the original role assignments. Having in mind that the new semantic frames are related to a subclass of verbs, they could be re-assigned for the whole class and thus facilitate the annotation process. In 64 cases (18.39%) the roles were changed because of passivized sentences in which the mapping rules have to be changed, or because of the attributive use of the participles.

The resulting semantic role sets show a greater variety in the verb.body class (12), an average variety in verb-consumption class (5) and hardly any variety in the verb.weather class (2).

In the verb.body class the subject in the frames is predominantly agent, sometimes experiencer, and rarely recipient or source; the direct object is mostly theme, sometimes patient or manner, and rarely communication, source, stimulus.

In the verb.consumption class the number of semantic frames is smaller, but highly varied. Here the subject can present the semantic roles of agent, asset, theme and pivot, while the direct object can present the semantic roles of patient, goal, predicate, asset and theme.

In the verb.weather class the subject is either theme or cause. The direct object is theme.

The numbers presented here show, on the one hand, that our approach saves manual work through the inheritance strategy, but it also ensures consistency of the semantic role frames assigned to the subclasses of verbs within the general verb classes.

6 Conclusion

In this paper we presented a procedure for semantic role annotation in a treebank, which has been already annotated with valencies and sense information. Since the mapping of the valencies and senses to the Princeton WordNet was performed manually and took some time, our procedure has not used directly the valency frames with the semantic roles within them, but rather the semantic classes with incremental assignment of the semantic roles.

The WordNet semantic classes have been transferred from English to Bulgarian via the WordNet mapping of verbs. The selected procedure helped us to minimize the post-editing of the automatically assigned semantic roles, since: the assignment process is incremental and allows for the gradual addition and further specification of the semantic role annotation; the change of the frames is distributed via the valency chains and syntactic labels into the data.

As future work we envisage the completion of semantic role refinement over all the remaining verb semantic classes; handling passivization and idiosyncratic cases.

Acknowledgements

This research has received support by the EC's FP7 (FP7/2007-2013) project under grant agreement number 610516: "QTLeap: Quality Translation by Deep Language Engineering Approaches." We would like to thank the three reviewers for the valuable feedback on our initial paper version. All errors remain our own.

References

[1] Fernando Gomez. Linking wordnet verb classes to semantic interpretation. In *In Proceedings of the COLING-ACL Workshop on the Usage of WordNet in NLP Systems*, pages 58–64, 1998.

[2] Arriola J. I. Aldezabal, Aranzabe M. and Díaz de Ilarraza A. A methodology for the semiautomatic annotation of Epec-rolsem, a Basque corpus labeled at predicative level following the Propbank-Verb Net model. In *Technical report*, 2013.

[3] Anna Korhonen. Assigning verbs to semantic classes via wordnet. In *Proceedings of the 2002 Workshop on Building and Using Semantic Networks - Volume 11*, SEMANET '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[4] Beth Levin. *English verb classes and alternations: a preliminary investigation*. Chicago: The University of Chicago Press, 1993.

[5] Petya Osenova, Kiril Simov, Laska Laskova, and Stanislava Kancheva. A treebank-driven creation of an ontovalence verb lexicon for Bulgarian. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2636–2640, Istanbul, Turkey, 2012. LREC 2012.

[6] Alexander Popov, Stanislava Kancheva, Svetlomira Manova, Ivaylo Radev, Kiril Simov, and Petya Osenova. The sense annotation of BulTreeBank. In *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)*, pages 127–136, Tuebingen, Germany, 2014. TLT 2014.

Universal Dependencies for Danish

Anders Johannsen, Héctor Martínez Alonso and Barbara Plank

Center for Language Technology
University of Copenhagen, Denmark
E-mail: {ajohannsen|alonso}@hum.ku.dk, bplank@cst.dk

Abstract

The Universal Dependencies (UD) project aims at developing treebank annotations consistent across many languages. In this paper, we present the conversion of the Copenhagen Dependency Treebank (CDT) into Universal Dependencies (UD). We describe the original CDT annotation and detail the mapping into the new UD formalism, which we accomplish by taking a *test-driven* approach. We present parsing experiments with both formalisms. Additionally, we quantitatively compare the resulting Danish UD treebank to the other languages available in the UD project (v1.2), discussing constructions that are specific to Danish. Our results show that the newly created Danish UD treebank is closely related to treebanks of typologically similar languages. However, parsing with the new treebank becomes more difficult, relative to the old formalism.

1 Introduction

The Universal Dependencies (UD) project[1] [15] is an on-going research effort that aims to facilitate multilingual and cross-lingual language technology. The UD project develops a dependency formalism that maximizes parallelism between languages, while allowing for language-specific extensions. In UD, content words are first-class citizens, and syntactic analyses that directly connect content words are preferred, whenever possible. The treebank annotation scheme grew out of three research related projects, namely the Stanford dependencies [7], the Google universal Part-of-Speech tag set [16] and the Interset interlingua for morphology [20]. The latest version of UD, v1.2, released November 2015, contains treebanks for 33 languages [14].

We introduce UD in Section 2.1, followed by a description of the somewhat atypical original annotation of the Copenhagen Dependency Treebank (CDT). Our conversion steps are described in Section 3. In Section 4 we assess the learnability of automatic parsers from the converted treebank and provide a quantitative evaluation of the resulting treebank when compared to the other UD languages.

[1] http://universaldependencies.github.io/docs/

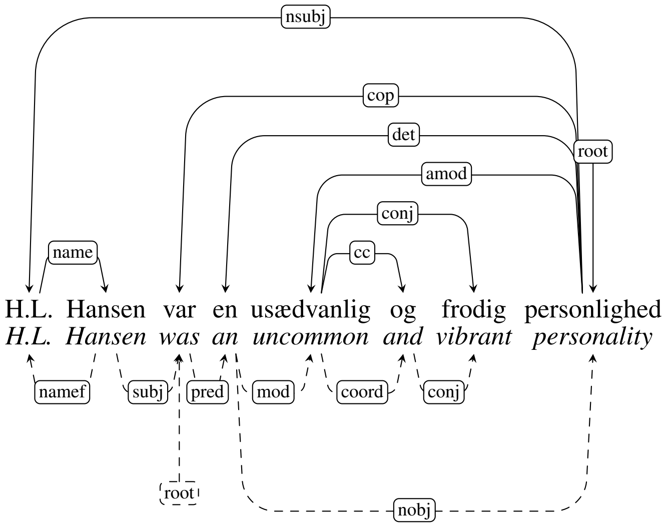

nsubj
cop
det
root
amod
conj
name
cc
H.L. Hansen var en usædvanlig og frodig personlighed
H.L. Hansen was an uncommon and vibrant personality
namef
subj
pred
mod
coord
conj
root
nobj

Figure 1: Dependency tree example. Above: UD, below: CDT (dashed).

2 Differences between UD and CDT

In this section we provide a brief overview on the principles of the UD formalism. For further details on syntactic and morphological annotation, we refer to [6, 13]. We then describe the design choices of CDT that are characteristically different from UD.

2.1 Universal Dependencies

The UD formalism has, roughly speaking, three driving principles:

1. **Content over function**: Content words are the heads of function words, e.g. lexical verbs are the head of periphrastic verb constructions, nouns are the heads of prepositional phrases, and attributes are the head of copula constructions.
2. **Head-first**: In spans where it is not immediately clear which element is the head (the content-over-function rule does not apply straightforwardly), UD takes a head-first approach: the first element in the span becomes the head, and the rest of the span elements attach to it. This applies mostly to coordinations, multiword expressions, and proper names.
3. **Single root attachment**: Each dependency tree has exactly one token directly dominated by the artificial root node. Other candidates for direct root attachment are instead attached to this root-dominated token.

An illustrative example is shown in Figure 2.1. Here, the copula is headed by the attribute *personlighed* (content over function). The proper name span *H.L.*

Hansen has the first element as head (head-first) and the tree has a single node dominated by the root. Apart from these three principles, UD imposes no further projectivity constraints, other than punctuation attachment must preserve projectivity. Further examples of annotation differences are given in the appendix (Figure 5).

UD uses a common set of 17 POS tags [13] and 40 syntactic relations [6].

2.2 Copenhagen Dependency Treebank

The Copenhagen Dependency Treebank (CDT) [11] consists of 5,512 sentences (about 100k tokens). The Danish source texts were collected and part-of-speech annotated by the PAROLE-DK project [10]. Although the CDT annotation scheme is very rich, it departs from all three UD principles listed in the previous section.

Perhaps the most salient difference is that CDT has determiners as heads. This is illustrated in Figure 2.1 (dashed), where the determiner *en* is the head of the noun *personlighed*. This analysis is known as Determiner Phrase (DP) analysis. While common in generative grammar frameworks [1, 9, 8, 17], it is very rarely implemented in dependency-based treebanks. To the best of our knowledge, the only other dependency treebank that uses DP analysis besides CDT is the Turin University Treebank [3].

In contrast to UD, dependencies in CDT follow a chain structure, resulting in trees with more levels. For instance, periphrastic verb constructions ("jeg *ville have kunnet* købe", "I *would have been able to* buy") in CDT are headed by the first auxiliary, and each following verb depends on the previous one. In our example in Figure 2.1, the copula is verb-headed. Coordinations are headed by the first conjunct, but the second conjunct is a dependent of the conjunction (cf. *frodig* depends on the conjunction *og*). This coordination structure deviates from the second UD principle, which specifies that all elements in a span attach to the first element. Finally, the CDT treebank contains several multi-rooted trees.

Moreover, CDT has no special part-of-speech tags for determiners. Many Danish determiners come with a homographic pronoun (e.g. '*min* jakke er **min**', '*my* jacket is **mine**'), and CDT provides the same tag, interpreted as a pronoun, for all forms. Thereby, the determiner-pronoun distinction is not recoverable at the part-of-speech level. Figure 2 shows three examples of noun phrase annotation with different specifiers from CDT. Like in English, Danish possessive constructions such 'Anna's parents' show complementary distributions with possessive determiners and receive the same dependency analysis, namely as heads of their following noun (Example b). The third example is a noun phrase complementing a pronoun, but it has the same structure as the second. Nevertheless, *ham* in Example c) is a case-marked pronoun that has no homographic determiner and will not be interpreted as determiner during the conversion process, unlike the determiner *de* in Example b) or *den* in Example c).

Similarly, CDT tags for verbs do not distinguish between lexical verbs and functional verbs, such as modals and auxiliaries. We apply tree-structure heuristics to disambiguate between verb-auxiliary and pronoun-determiner (cf. Section 3).

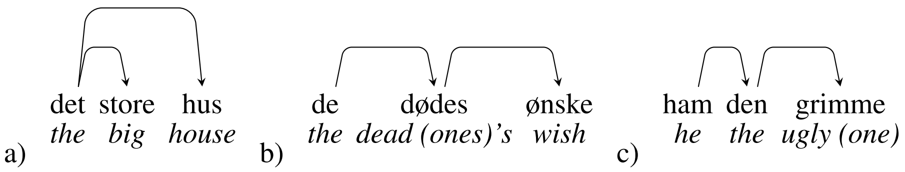


Figure 2: CDT treatment of different noun specifiers.

Our starting point for CDT is the data distributed in the CoNLL 2006 multilingual-parsing shared task [4]. We keep the same test set (322 sentences), but add a fixed development set of the same size by randomly sampling from the training set. All three data sets are disjoint. Prior to conversion, we added some missing lemmas and manually enforced single-rootness. The treebank consists of 100,777 tokens in 5,512 sentences with an average sentence length of 18.3 tokens.

3 Conversion

Inspired by best practices in software development, we take a *test-driven* approach to the tree-bank conversion. The most direct application of this methodology would be to establish a set of reference annotations and compare them to the result of running the conversion procedure. If they are not equal, the test "fails". However, this binary setup is not sufficient for larger conversions which are the result of chained application of many smaller conversion procedures or steps: the individual steps should be tested as well.

After each step we therefore automatically calculate a series of quality measures with respect to the reference set, such as labeled and unlabeled attachment score (LAS/UAS), tree-consistency (weak connectedness, single-rootness), and other indicators like number of non-projective edges, and average number of transformed edges.

Test bench The reference set contains 28 sentences, randomly sampled from the CDT treebank and annotated manually from scratch using the UD guidelines. The reference set has a *goal* subset of 17 sentences, where we expect the conversion process to obtain a LAS of 100% (test passed). These sentences exhibit all the basic syntactic phenomena addressed by the conversion steps listed in Table 1. The remaining 11 sentences in the reference set contain more rare phenomena like fragments of compounds, coordinated applications of several prepositions to the same noun ('*on and under* a tree'), or clausal complement labels like *csubj*. We use the whole reference set to measure the overall quality of the conversion and to provide directions for future improvements. The purpose of the goal subset is to test that the current conversion works predictably. If in the future we decide that a syntactic phenomenon like coordinated prepositions should fall within the scope of the conversion, we simply move sentences with this phenomenon to the goal subset. The resulting LAS for the goal subset is 100%, whereas for the overall reference set, the converted treebank scores 86.44% LAS and 89.54% UAS.

The conversion tool is implemented in Python as a sequence of rewrite operations on a graph structure. The conversion framework, which is treebank-independent, is available for download.[2] Table 1 shows the conversion steps and their score on the reference set. The following section describes the rewrite operations.

Rewrite operations The treebank conversion is the result of 18 sequentially applied rewrite operations. These fall into four broad categories. We first apply global operations involving conjunctions, then do local operations involving nouns, followed by operations that are verb-centric. During the conversion, we use part-of-speech information from the CDT. As one of the last steps, "Map POS and feats", features and dependency relations are mapped into UD labels. Finally, certain multi-word units (MWU), which in CDT appear as one token (e.g. 'i dag', lit. 'in day", 'today'), are split into their component tokens.

1. **Conjunction-centric (C)** The first group of operations involves flattening coordinating conjunction chains and applying the head-first principle.
2. **Noun-centric (N)** This group of operations rewrites DP analyses, making nouns heads. In particular, it implements rewriting of proper names, switching the headedness of determiners and possessives, as well as making adpositions case-markers of the content noun. We disambiguate determiners and pronouns according to their form and dependency relations. Specifically, an ambiguous pronoun becomes a determiner if it introduces a noun.
3. **Verb-centric (V)** The rewrite operations for verbs mainly involves flattening verb chains and making them content-headed, identifying content heads for copula constructions and making adpositions clause markers for their introduced verbs. We disambiguate verbs in auxiliaries and content verbs according to their form and dependency relations, namely by determining whether a verb belongs to the closed class of functional verbs and it introduces a lexical verb.
4. **Label-centric (L)** This group contains mappings and heuristics for relabeling of POS, morphological features and dependency relations. Most POS and features are obtained from Interset [20], while other traits (determiner, auxiliary, copula) are calculated from edge properties. The Danish UD dependency relation inventory comprises the standard UD inventory, plus three language-specific labels, namely *nmod:loc*, *nmod:tmod* and *nmod:poss*.

We observe how UAS and LAS increase monotonically from the first step to the last. We cannot say the same about projectivity, because the average non-projectivity increases after e.g. reattaching conjunctions and copulas. The last step

[2] https://github.com/andersjo/ud-test-driven-conversion

			Changes per sent.		Scores	
Conversion step		Non-projectivity	Labeled	Unlabeled	UAS	LAS
–	Identity transform	0.35	0.00	0.00	30.45	5.30
–	Preprocess	0.35	0.00	0.00	30.45	5.30
C	Discourse conjunctions	0.35	0.18	0.18	32.41	6.61
C	Switch sconj headness	0.35	0.00	0.00	32.41	6.61
C	Modify conjunctions	0.41	0.71	0.35	33.56	8.90
C	Switch clause relating element head	0.41	0.12	0.12	33.98	8.90
N	Proper names head first	0.41	0.59	0.59	37.72	12.19
N	Switch possessive head	0.41	0.59	0.59	39.27	13.42
N	Switch article head	0.41	2.88	2.88	58.15	28.57
V	Switch particle head	0.41	0.00	0.00	58.15	28.57
N	Switch preposition head	0.41	3.06	3.06	82.82	51.29
V	Infinite verb chains	0.41	0.00	0.00	82.82	51.29
V	Verb chains to content head	0.12	1.06	1.06	89.42	53.50
V	Copula to content head	0.29	1.12	1.12	98.17	56.25
–	Punct and subordination	0.29	0.29	0.29	100.00	56.25
L	Map POS and feats	0.29	0.00	0.00	100.00	56.25
L	Map deprels	0.29	6.06	0.00	100.00	100.00
–	Prepositions as leaves	0.29	0.00	0.00	100.00	100.00
–	Split multi-word units	–	–	–	–	–

Table 1: Conversion statistics on the *goal* reference annotations. Lab and unlab Δ: mean number of labeled or unlabeled changes.

involves re-tokenization and can only be applied when the rest of the tree structure has been reassigned. Therefore no scores are provided for it in Table 1. The operations that have a larger impact in terms of how many edges are reattached are “switch article head” and “switch preposition head”, which reattach determiners and prepositions respectively. Moving determiners and prepositions from functional heads to leaves in the tree has a large impact on the overall structure of the trees, because their dependents must also be reattached.

4 Evaluation

Parsing We train two state-of-the-art graph-based dependency parsers, MST (2nd order, non-proj) and Mate [12, 2] on the original (CDT) and converted UD-Danish data. The results in Table 2 show a 4-5% accuracy drop when parsing UD-Danish with standard features. This is not surprising, as CDT and UD-Danish are now quite different treebanks. In fact, the attachment of 65% of the edges changed during the conversion. In contrast to our results, Pyysalo et al. [18] observed only a minor drop in performance (0.5%) on Finnish. However, their original annotation is based on Stanford dependencies and is thus closer to UD than CDT.

The MST parser’s performance drop on labeled accuracy (compared unlabeled accuracy) is remarkable. Both parsers are second-order, but Mate has more context features. To get an intuition about the difficulty of predicting dependency labels for CDT and UD, respectively, we train a simple linear-chain CRF model which

	Mate			MST		
	LAS	UAS	LA	LAS	UAS	LA
CDT DEV	85.20	89.38	90.83	84.59	89.46	90.61
CDT TEST	84.38	88.70	90.17	84.11	89.44	90.69
UD-DANISH DEV	81.87	84.51	92.10	65.87	81.57	75.71
UD-DANISH TEST	81.56	84.64	92.00	63.87	80.91	74.54

Table 2: Parsing accuracy including punctuation.

for each token outputs the label of its head relation. Only the word itself and the universal part-of-speech tag is used as input. The model obtains slightly higher accuracies for predicting UD labels (88.21% vs 87.97% on the test set). So, in the absence of structural information, there seems to be little difference in the predictability of labels in UD and CDT.

Similarity with other UD treebank We estimate the similarity with other UD treebanksby comparing several distributions, i.e. distributions over labels, over POS, and over labeled head-dependent head triples. We compare Danish to three sets of languages; **Scandinavian** (no,sv), **Germanic** languages (de,en,nl,no,sv) and **all** languages.[3] Due to space restrictions, we here mainly focus on the comparison with Norwegian and Swedish (cf. Figure 3a). Danish has fewer `det` relations than the other two Scandivinavian languages, but even fewer than the average language in UD. We attribute this difference between the Scandinavian languages and the rest to their nominal definite inflection pattern [5].

More surprisingly, we observe that Danish stands out in the amount of *punct* relations. Examining Figure 4, we observe that punctuations have far longer average dependency length for punctuations than the UD treebanks as a whole. This difference might be a result of the relatively high number of punctuation symbols, as well as the reattachment operations that attach punctuation far from the dependent to avoid crossing edges. We observe a similar pattern for average head distance in coordinations, which might also be a result of the heuristics applied in the coordination.

5 Conclusions and future work

We presented a test-driven conversion of the Copenhagen Dependency Treebank (CDT) into Universal Dependencies (UD).

Conversion to UD is an ongoing process, as the standard converges across languages. We expect to revise several aspects of the treebank for a future release: 1) a homogenous analysis of proper-name headedness in the presence of other nominal complements ('the newspaper *The New York Times*'); 2) a semi-manual validation

[3] We compare with the over UD treebanks from version 1.2, released November 15th, 2015.

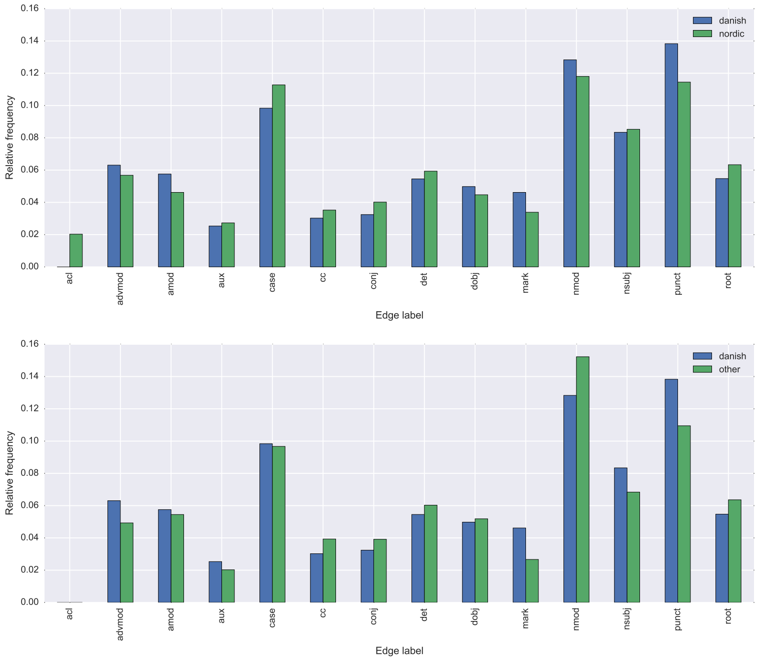


Figure 3: Dependency labels distribution comparison for Danish vs. Nordic Swedish and Norwegian (above), and for Danish vs. all languages (below).

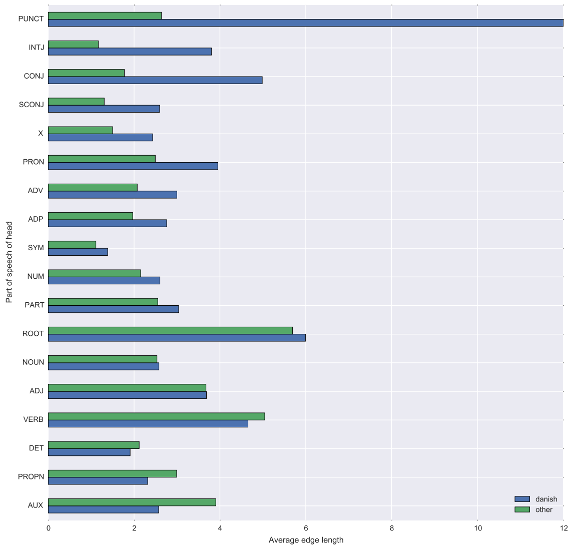


Figure 4: Average distance to head by part of speech of the dependent, compared between Danish and the average of all other UD treebanks.

of the *aux/auxpass* labels for periphrastic movement verbs, because Danish movement verbs like *ankomme* (‘arrive’) use the verb *være* (‘be’) as auxiliary, and it should not be treated as *auxpass*; 3) a revision of the re-attachment of coordinating conjunctions and punctuations to control for distance to head node, and; 4) a revision of the labels for clausal complements like ccomp or csubj. This step is arguably the most difficult to automate, and might require an annotation task. Silviera & Manning [19] discuss in more details the issues of labeling phrasal and clausal relations on one layer in dependency analyses.

References

[1] Steven Paul Abney. *The English noun phrase in its sentential aspect*. PhD thesis, Massachusetts Institute of Technology, 1987.

[2] Bernd Bohnet. Top accuracy and fast dependency parsing is not a contradiction. In *COLING*, 2010.

[3] Cristina Bosco, Vincenzo Lombardo, Daniela Vassallo, and Leonardo Lesmo. Building a treebank for Italian: a data-driven annotation schema. In *LREC*, 2000.

[4] Sabine Buchholz and Erwin Marsi. Conll-x shared task on multilingual dependency parsing. In *CoNLL*, pages 149–164, 2006.

[5] Östen Dahl. Definite articles in scandinavian: Competing grammaticalization processes in standard and non-standard varieties. *Dialectology Meets Typology: Dialect grammar from a cross-linguistic perspective*, pages 147–180, 2004.

[6] Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of LREC*, pages 4585–4592, 2014.

[7] Marie-Catherine de Marneffe and Chris Manning. Stanford typed dependencies manual. In *Technical report*, 2008.

[8] Jorge Hankamer and Line Mikkelsen. A morphological analysis of definite nouns in danish. *Journal of Germanic Linguistics*, 14(02):137–175, 2002.

[9] Richard A Hudson. *Word grammar*. Blackwell Oxford, 1984.

[10] Britt Keson. Det danske morfosyntaktisk taggede PAROLE-korpus. Technical report, DSL, 2004.

[11] M.T. Kromann and S.K. Lynge. Danish Dependency Treebank v. 1.0. Department of Computational Linguistics, Copenhagen Business School., 2004.

[12] Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *ACL*, 2005.

[13] Joakim Nivre. Towards a universal grammar for natural language processing. In *Computational Linguistics and Intelligent Text Processing*, 2015.

[14] Joakim Nivre, Željko Agić, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Cristina Bosco, Sam Bowman, Giuseppe G. A. Celano, Miriam Connor, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Daniel Galbraith, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Berta Gonzales, Bruno Guillaume, Jan Hajič, Dag Haug, Radu Ion, Elena Irimia, Anders Johannsen, Hiroshi Kanayama, Jenna Kanerva, Simon Krek, Veronika Laippala, Alessandro Lenci, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Shunsuke Mori, Hanna Nurmi, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Prokopis Prokopidis, Sampo Pyysalo, Loganathan Ramasamy, Rudolf Rosa, Shadi Saleh, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Kiril Simov, Aaron Smith, Jan Štěpánek, Alane Suhr, Zsolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Sumire Uematsu, Larraitz Uria, Viktor Varga, Veronika Vincze, Zdeněk Žabokrtský, Daniel Zeman, and Hanzhi Zhu. Universal dependencies 1.2, 2015. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

[15] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal dependencies v1: A multilingual treebank collection. *LREC*, 2016, under review.

[16] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. CoRR abs/1104.2086, 2011.

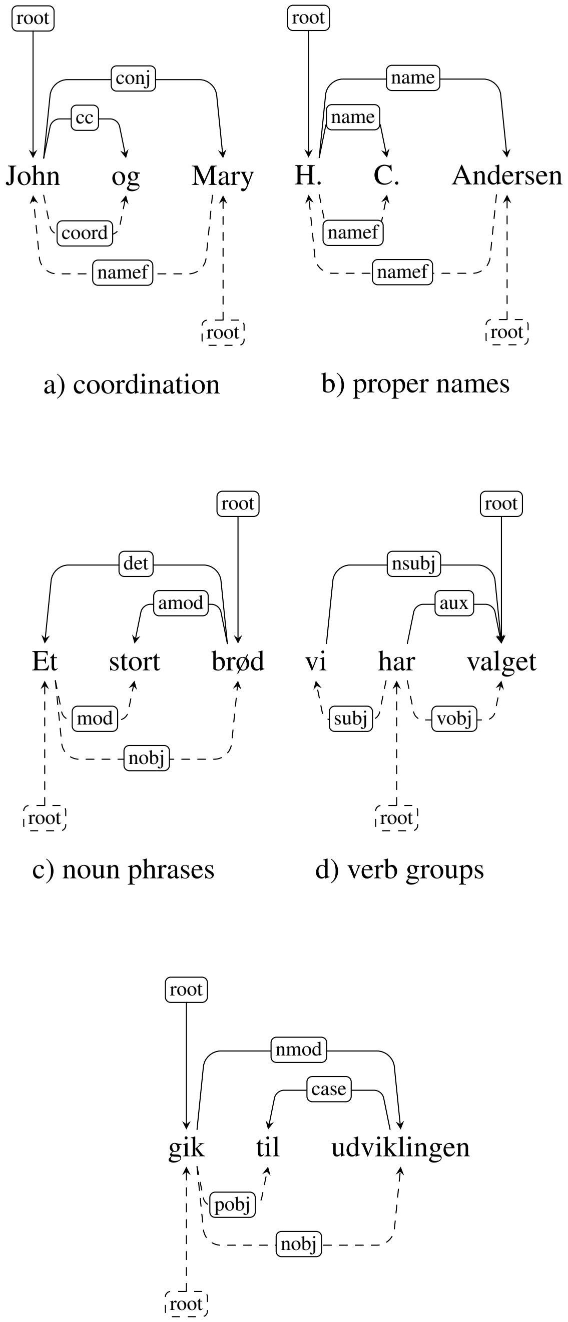
[17] Jeffrey Punske. Functional structure inside nominal phrases. *The Routledge Handbook of Syntax*, page 65, 2014.

[18] Sampo Pyysalo, Jenna Kanerva, Anna Missilä, Veronika Laippala, and Filip Ginter. Universal Dependencies for Finnish. In *Nordic Conference of Computational Linguistics NODALIDA 2015*, page 163, 2015.

[19] Natalia Silveira and Christopher Manning. Does universal dependencies need a parsing representation? an investigation of english. *Depling 2015*, page 310, 2015.

[20] Daniel Zeman. Reusable tagset conversion using tagset drivers. In *LREC*, 2008.

Appendix



a) coordination

b) proper names

c) noun phrases

d) verb groups

e) prepositions (in prepositional phrases or infinite verb phrases)

Figure 5: Example of annotation differences; CDT scheme (dashed); UD (solid).

Supporting LFG Parsing with Dependency Parsing

Adam Przepiórkowski[1,2] and Alina Wróblewska[1]

[1]Institute of Computer Science, Polish Academy of Sciences
[2]University of Warsaw
E-mail: {adamp|alina}@ipipan.waw.pl

Abstract

In order to increase the coverage of the Polish LFG grammar, a novel method of combining grammar-based and data-driven parsers is proposed consisting in 1) augmenting the LFG grammar with so-called FRAGMENT rules which make it possible to obtain substructures for parsable fragments of sentences, 2) composition of such FRAGMENT substructures into a full f-structure on the basis of dependency relations proposed by an independent data-driven parser, with 2a) modification of the internal structure of FRAGMENT substructures only if absolutely necessary for independent (LFG-theoretical) reasons and 2b) modification of dependency relations in accordance with the naming and grammatical conventions of the LFG grammar.

1 Introduction

Dependency parsers trained on large treebanks have obvious advantages over parsers relying on manually-constructed grammars adhering to linguistic formalisms such as Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag 1987, 1994) or Lexical Functional Grammar (LFG; Bresnan 1982, 2001; Dalrymple 2001). Data-driven parsers have good coverage and produce a single (i.e. best, according to some metric) parse for a sentence, while grammar-based parsers – unless they are supported by pre- and post-processing heuristics and/or some data-driven training – usually have poor coverage and often produce numerous parses for the sentences that they do cover. On the other hand, the resulting analyses based on manually-constructed grammars are often much deeper syntactically and may also contain semantic information often missing from data-driven dependency parsers, which concentrate on grammatical functions (subject, object, etc.). Hence, some work has been devoted to combining the two kinds of approaches (see Section 6 below).

The aim of this paper is to propose a novel way of combining grammar-based and data-driven dependency parsing. If the grammar-based parser produces a single parse, nothing needs to be done. Otherwise, there are two possibilities: the grammar-based parser produces many parses, or it finds none. In both cases, an independently constructed data-driven dependency parser is employed to parse the same sentence and produce a single dependency tree. Trivially, this dependency tree may be used to disambiguate between parses produced by the grammar-based parser. Less trivially, in case the grammar-based parser finds no complete parse but manages to construct representations for fragments of the sentence, the dependency tree may be used to glue these representations together into a single parse. This paper presents an implementation of such a less trivial scenario.

The method proposed here is tested with the Polish LFG grammar, POLFIE (`http://zil.ipipan.waw.pl/LFG`; Patejuk and Przepiórkowski 2012, 2014), which suffers from the usual coverage problems of manually-constructed grammars. The solution consists in applying the following stages of processing. First, the text is parsed with pure POLFIE. Second, unparsed sentences are processed with POLFIE augmented with a so-called FRAGMENT sub-grammar, which makes it possible to construct an artificial parse containing f-structures corresponding to parsed fragments. Finally, such partial f-structures are composed into a single coherent f-structure using dependency relations in the dependency tree found by the data-driven parser.

There are two assumptions behind this procedure that should be made explicit. First, while LFG analyses are represented by c- and f-structures, only f-structures are modified in the current approach, while c-structures are discarded as much less important for further (esp. semantic) processing. Second, the internal structure of sub-f-structures produced by the augmented grammar should – in principle – not be modified. However, some modification is necessary in order to apply well-formedness LFG principles, i.e. completeness, coherence, and uniqueness (see Dalrymple 2001 or Bresnan 2001).

2 LFG-based partial parsing

Lexical Functional Grammar focuses on syntax and its relations with morphology, semantics and pragmatics. The syntactic dimension of a sentence contains two main (parallel) representations – a constituent structure (c-structure; essentially, a context-free phrase structure) and a functional structure (f-structure; a finite set of attribute-value pairs that encode functional properties of a sentence).

Handcrafted LFG grammars are typically implemented within the Xerox Linguistic Environment (XLE; Maxwell III and Kaplan 1993; Crouch *et al.* 2011). Apart from parsing complete sentences, XLE provides a mechanism for parsing possibly incorrect sentences: if a string of tokens cannot be parsed with standard rules of a grammar (i.e. correct c- and f-structures cannot be assigned to this string), it is reparsed with a grammar augmented by so-called FRAGMENT rules, and a sequence of well-formed partial structures specified by the augmented grammar is produced, as in Figure 1. The final set of such “FRAGMENT parses” is selected in a way that minimises the number of partial structures (i.e. parses of larger

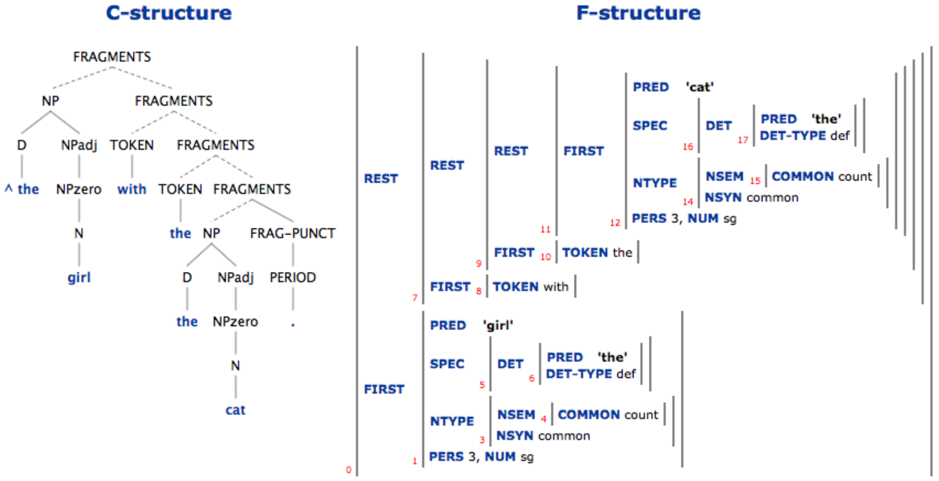


Figure 1: The FRAGMENT LFG analysis of 'The girl with the the cat.'

chunks are preferred to those of smaller chunks). Furthermore, it is possible that some words cannot be recognised and interpreted morphosyntactically, since they are misspelled or are not represented in the lexicon. Such words, as well as words which are not covered by any grammatical rules in the current parse, are encoded as unparsed TOKENs in f-structures. The FRAGMENT parses are assigned the root category FRAGMENTS. They are encoded with FRAGMENT and TOKEN nodes in c-structures and FIRST and REST functions in f-structures.

This view of FRAGMENT parsing seems to be language-independent: FRAGMENTS are combined in a linear manner consistent with the order of tokens – the sub-f-structure of the FRAGMENT which is first on the list is a value of the attribute FIRST and the sub-f-structure of all other FRAGMENTS on the list is a value of the attribute REST. However, what kinds of phrases or tokens can constitute FRAGMENTS is decided by the designers of the grammar, so the resulting FRAGMENT grammar is to some extent language-dependent.

For English, the combination of full and fragment parsing techniques allows for achieving 100% grammar coverage on unseen data (cf. Riezler *et al.* 2002).

3 Augmented grammar

The Polish LFG grammar POLFIE consists of 65 large rules (i.e. with disjunctive right-hand sides), which compile into a finite-state automaton with 479 states and 1041 arcs. There are no guessers – neither for the output of the morphological analyser (i.e. all analyses output by the morphological analyser build the lexicon for a particular sentence), nor for words unrecognised by the analyser. Since writing grammar rules is a very-time consuming process, there are still many constructions that are not defined yet in the grammar. POLFIE covers only about 40% of a representative corpus of Polish (cf. the initial row in Table 1).

Table 1: Test coverage on 20K sentences from the manually annotated part of National Corpus of Polish (`http://nkjp.pl/`; Przepiórkowski *et al.* 2012) parsed with POLFIE

grammar	parsed sentences	unparsed sentences	errors and time out
POLFIE	8364 (42%)	8181 (41%)	3455 (17%)
FRAGMENT grammar	11349 (57%)	0	8648 (43%)
FRAGMENT grammar with OT-marks and pruning	18909 (95%)	0	1091 (5%)

In order to increase the coverage of the Polish LFG grammar, the technique of partial parsing described in the previous section is applied (cf. the penultimate row in Table 1). The procedure of partial parsing makes it possible to parse a larger number of sentences which otherwise receive no analysis. However, in contrast to English, where partial parsing is used to parse incorrect sentences, Polish sentences with FRAGMENT parses are not necessarily incorrect. Many of them are well-formed but contain linguistic phenomena for which POLFIE rules have not been defined yet. FRAGMENT analyses of such sentences are candidates for improvement with the proposed method.

A quantitative analysis shows that the augmented grammar produces a huge number of analyses for some sentences. In order to limit the number of parses the augmented grammar is extended with some optimality marks.[1] Furthermore, in order to reduce the number of memory and time out errors, the XLE mechanism of pruning c-structures before processing f-structure constraints is employed.[2] The statistics of parsing with the extended version of the LFG grammar is presented in the final row in Table 1.

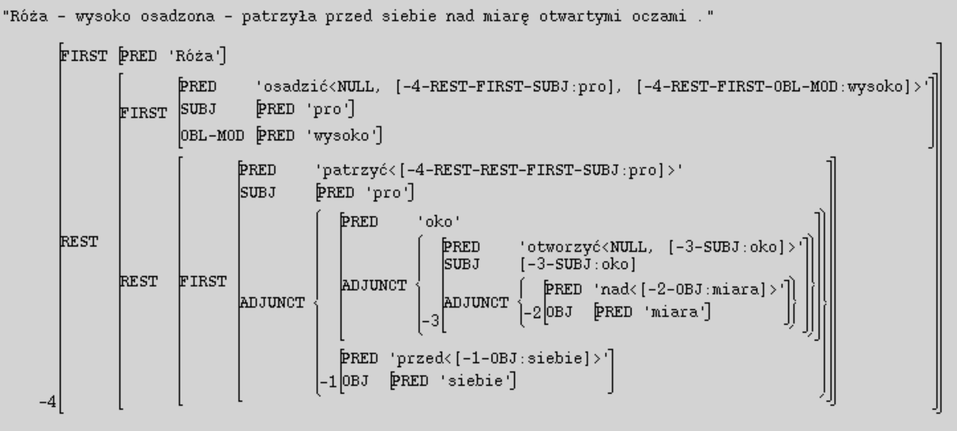
4 Recomposition of FRAGMENT sub-f-structures

The recomposition method only applies to LFG analyses with the root category FRAGMENTS. Since only f-structures are currently the subject of modification, the set of all LFG analyses output for a sentence is restricted to the set of analyses with unique f-structures. The main idea behind the method consists in the recomposition of FRAGMENT substructures in f-structures in accordance with dependency relations between their highest PRED attributes.

[1] Optimality theory marks (i.e. OT-marks) are preference and dispreference marks which are used to rank grammar rules, templates, and lexical entries. The most preferable grammar rules and templates can be applied and the most preferable lexical entries can be selected, e.g. in order to resolve ambiguity. While there are no OT-marks in the publicly avaliable POLFIE version used in the current experiments, we augmented the FRAGMENT rules with some OT-marks and defined their ranking.

[2] XLE documentation on pruning: `http://www2.parc.com/isl/groups/nltt/xle/doc/xle.html#SEC14J`.

A FRAGMENT f-structure consists of sub-f-structures connected with attributes FIRST and REST (see Figure 2). New relations between these sub-f-structures (see Figure 3) are determined on the basis of the corresponding dependency structure (see Figure 4).

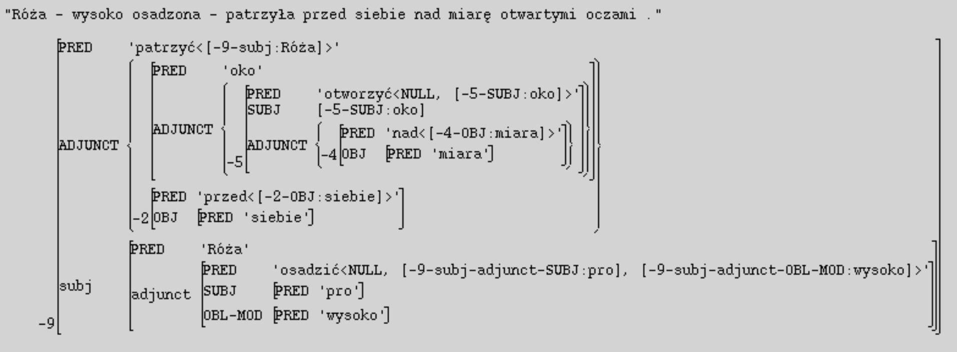


```
"Róża - wysoko osadzona - patrzyła przed siebie nad miarę otwartymi oczami ."

    FIRST PRED 'Róża'
                PRED    'osadzić<NULL, [-4-REST-FIRST-SUBJ:pro], [-4-REST-FIRST-OBL-MOD:wysoko]>'
          FIRST SUBJ    PRED 'pro'
                OBL-MOD PRED 'wysoko'
                      PRED    'patrzyć<[-4-REST-REST-FIRST-SUBJ:pro]>'
                      SUBJ    PRED 'pro'
                                 PRED    'oko'
    REST                                   PRED    'otworzyć<NULL, [-3-SUBJ:oko]>'
                                           SUBJ    [-3-SUBJ:oko]
          REST  FIRST                    
                                 ADJUNCT           PRED 'nad<[-2-OBJ:miara]>'
                      ADJUNCT              ADJUNCT
                                        -3       -2 OBJ  PRED 'miara'

                                 PRED 'przed<[-1-OBJ:siebie]>'
                              -1 OBJ  PRED 'siebie'
 -4
```

Figure 2: The FRAGMENT parse of *Róża – wysoko osadzona – patrzyła przed siebie nad miarę otwartymi oczami.* (Eng. ‘Rose – placed highly – was looking straight ahead with her eyes open too widely.’)



```
"Róża - wysoko osadzona - patrzyła przed siebie nad miarę otwartymi oczami ."

    PRED     'patrzyć<[-9-subj:Róża]>'
                PRED    'oko'
                          PRED    'otworzyć<NULL, [-5-SUBJ:oko]>'
                          SUBJ    [-5-SUBJ:oko]
                ADJUNCT
    ADJUNCT                               PRED 'nad<[-4-OBJ:miara]>'
                          ADJUNCT
                       -5              -4 OBJ  PRED 'miara'

                PRED 'przed<[-2-OBJ:siebie]>'
             -2 OBJ  PRED 'siebie'
             PRED    'Róża'
                     PRED    'osadzić<NULL, [-9-subj-adjunct-SUBJ:pro], [-9-subj-adjunct-OBL-MOD:wysoko]>'
    subj     adjunct SUBJ    PRED 'pro'
                     OBL-MOD PRED 'wysoko'
 -9
```

Figure 3: The recomposed f-structure of the FRAGMENT f-structure of Figure 2; dependency labels used as new attributes are written in lower case

Dependency structures are generated with an external data-driven dependency parser MATE (Bohnet, 2010) trained on the Polish dependency treebank (Wróblewska, 2014). Although the parsing quality of short and simple sentences with manually annotated tokens is relatively high (87.2 LAS and 92.7 UAS), the parsing quality of complex sentences with semi-automatically annotated tokens is significantly lower (70.3 LAS and 76.0 UAS) – see Wróblewska 2014 for details. As dependency trees may contain errors, the internal structures of rule-based sub-f-structures are not modified when they disagree with the dependency tree, unless

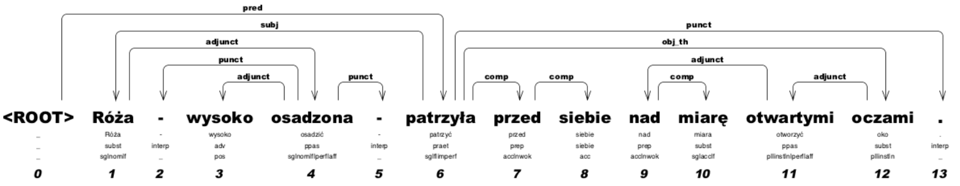
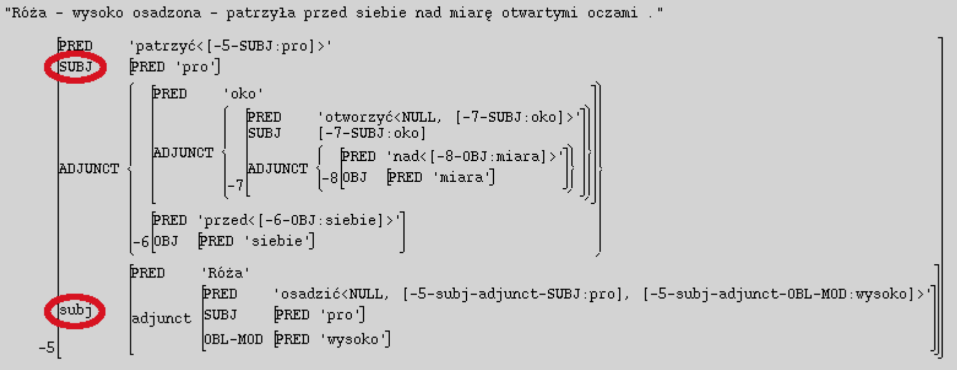


Figure 4: Dependency tree of *Róża – wysoko osadzona – patrzyła przed siebie nad miarę otwartymi oczami.* (Eng. 'Rose – placed highly – was looking straight ahead with her eyes open too widely.')

such a modification is essential for constructing a coherent f-structure for the whole sentence.

An essential modification of sub-f-structures includes removal of sub-f-structure for a pro-drop pronoun with the SUBJ (or OBJ) function, if one of FRAGMENT substructures is annotated as SUBJ (or OBJ) in the modified f-structure, in order to avoid f-structures with double subject. An example of an incoherent f-structure is given in Figure 5.



```
"Róża - wysoko osadzona - patrzyła przed siebie nad miarę otwartymi oczami ."

   PRED      'patrzyć<[-5-SUBJ:pro]>'
   SUBJ      [PRED 'pro']
                 PRED    'oko'
                          PRED     'otworzyć<NULL, [-7-SUBJ:oko]>'
                          SUBJ     [-7-SUBJ:oko]
                 ADJUNCT {
   ADJUNCT  {                ADJUNCT { PRED 'nad<[-8-OBJ:miara]>'
                      -7              -8 OBJ  [PRED 'miara']

                 PRED 'przed<[-6-OBJ:siebie]>'
              -6 OBJ  [PRED 'siebie']
             PRED     'Róża'
                      PRED     'osadzić<NULL, [-5-subj-adjunct-SUBJ:pro], [-5-subj-adjunct-OBL-MOD:wysoko]>'
   subj      adjunct  SUBJ     [PRED 'pro']
                      OBL-MOD  [PRED 'wysoko']
-5
```

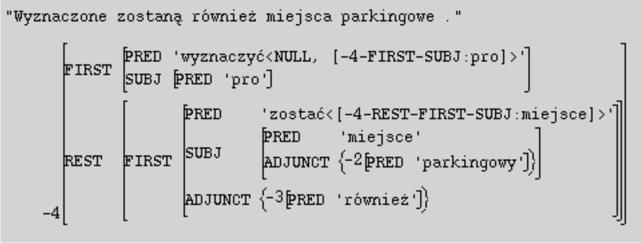
Figure 5: The incoherent f-structure glued from sub-f-structures in Figure 2

Rules converting dependency relations into an f-structure must convert dependency labels into f-structure attributes, e.g. – trivially – the *subj* dependency label is converted into the SUBJ attribute. Less trivially, as some linguistic phenomena (e.g. passive voice, analytical predicative constructions with *to*[3]) are treated differently in POLFIE and by the dependency parser, conversion rules must also perform some regular restructuring.

In passive constructions encoded in POLFIE f-structures, the participle is annotated as an XCOMP-PRED dependent of the auxiliary verb *zostać*. By contrast, in dependency structures, the participle functions as the governor of the auxiliary *zostać*. For example, the sentence *Wyznaczone zostaną również miejsca*

[3] The predicative *to* is a governor of an auxiliary verb form in dependency trees. In f-structures, in turn, the auxiliary is not encoded as a function with the f-structure value but as a value of the attribute TENSE incorporated into the f-structure of *to*.

parkingowe. (Eng. ‘Parking spaces will also be designated.’), when parsed by the LFG grammar augmented with FRAGMENT rules, receives the f-structure given in Figure 6. In order to glue the two FRAGMENT sub-f-structures, the same



```
"Wyznaczone zostaną również miejsca parkingowe ."

    FIRST  [PRED 'wyznaczyć<NULL, [-4-FIRST-SUBJ:pro]>']
           [SUBJ [PRED 'pro']                           ]

                  [PRED      'zostać<[-4-REST-FIRST-SUBJ:miejsce]>'
                  [          [PRED     'miejsce'                  ]
    REST   FIRST  [SUBJ      [ADJUNCT {-2[PRED 'parkingowy']}     ]
                  [
                  [ADJUNCT {-3[PRED 'również']}
-4
```

Figure 6: The FRAGMENT parse of *Wyznaczone zostaną również miejsca parkingowe.* (Eng. ‘Parking spaces will also be designated.’)

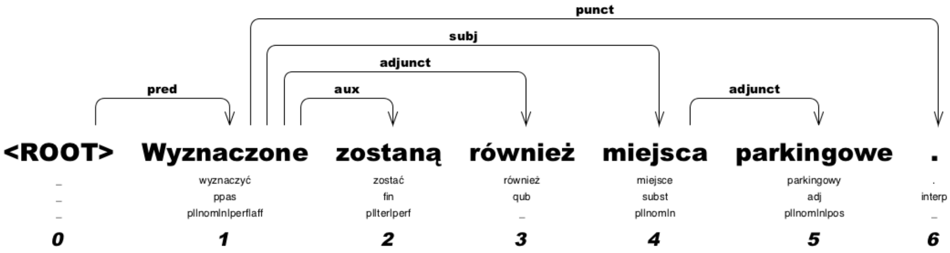
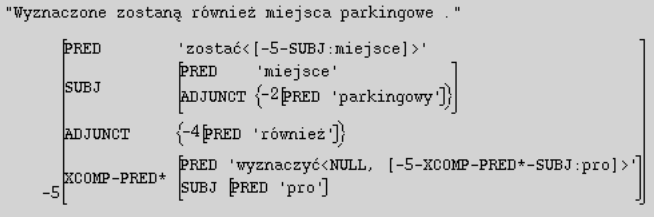


Figure 7: Dependency tree of *Wyznaczone zostaną również miejsca parkingowe.* (Eng. ‘Parking spaces will also be designated.’)

sentence is parsed with the dependency parser, and the resulting dependency tree (see Figure 7) is inspected for the presence of the two words corresponding to the top PRED values of the two fragments: *wyznaczone* ‘designated’ (cf. `PRED 'wyznaczyć<...>'` in Figure 6) and the auxiliary *zostaną* (`PRED 'zostać<...>'` in that figure). These two words are connected by an arc with the *aux* (auxiliary verb) dependency label, the governor’s part of speech is *ppas* (passive adjectival participle), and the dependent’s lemma is ZOSTAĆ, so this is a passive construction according to the dependency tree, and it is translated into the LFG analysis of passive adopted in POLFIE; the resulting f-structure is given in Figure 8.

If it is not possible to match any dependency label to an appropriate LFG grammatical function, the dependency label is left without any modification. The properly converted labels are marked with the asterisk (see XCOMP-PRED in Figure 8), in order to retain information about newly introduced attributes, not generated by the rules of the LFG grammar.



```
"Wyznaczone zostaną również miejsca parkingowe ."

    PRED          'zostać<[-5-SUBJ:miejsce]>'
                  PRED     'miejsce'
    SUBJ          ADJUNCT {-2[PRED 'parkingowy']}

    ADJUNCT       {-4[PRED 'również']}

                  PRED 'wyznaczyć<NULL, [-5-XCOMP-PRED*-SUBJ:pro]>'
    XCOMP-PRED*   SUBJ [PRED 'pro']
 -5
```

Figure 8: The recomposed f-structure of the FRAGMENT f-structure of Figure 6

5 Evaluation

There is no gold standard data that could be used for the evaluation of the proposed gluing procedure. In this very preliminary evaluation, 30 FRAGMENT analyses were randomly selected from the set of all those FRAGMENT analyses which have only one solution. Sub-f-structures of these analyses were then manually glued into proper f-structures, and the resulting 30 f-structures were used as the gold standard.

As assumed, FRAGMENT analyses with only one solution were generated mostly for relatively short sentences. There are 6.9 tokens per sentence on average in the resulting test set. These sentences were not covered by the POLFIE grammar for various reasons, e.g. wrong punctuation, lack of rules for predicate-less sentences, reported speech, unconventional word order, or eliptical constructions.

In order to evaluate glued f-structures, some metrics inspired by dependency parsing metrics are defined: UAS – the percentage of FIRST sub-f-structures that are correctly unified with the final f-structure, LAS – the percentage of FIRST sub-f-structures that are correctly unified with the final f-structure as a value of a correct grammatical function, and LA – the percentage of FIRST sub-f-structures that are incorporated as a value of a correct grammatical function. Tested against the set of 30 manually composed f-structures, the following results obtain: 79.45% UAS, 72.6% LAS, and 78.08% LA.

Additionally, the choice of grammatical functions was evaluated using the usual measures: precision, recall, and f-measure (see Table 2). The repertoire of evaluated grammatical functions is not representative since many functions do not appear in test data. On the one hand, it is because of the simplicity of test data. On the other hand, it is because of relatively good treatment of some grammatical functions (e.g. XCOMP, POSS) in POLFIE; these functions only appear inside of sub-f-structures.

The results indicate that many sub-f-structures are glued with a correct grammatical function. In particular, sub-f-structures that function as the sentence predicate (i.e. the value of PRED for the whole sentence) or are values of ADJUNCT and OBJ attributes are correctly identified in most cases. The results also indicate what should be improved in the proposed gluing procedure.

Table 2: Precision, recall and f-measure of individual grammatical functions gluing sub-f-structures in the test FRAGMENT f-structures

grammatical function	number of occurrences	precision	recall	f-score
ADJUNCT	21	90%	86%	0.88
ADJUNCT-QT	9	100%	22%	0.36
SUBJ	3	33%	66%	0.44
OBJ	2	66%	100%	0.80
OBL-GEN	2	100%	50%	0.66
COMP	1	100%	100%	1.00
sentence predicate	27	96%	88%	0.92
coordination conjunct	6	54%	100%	0.71

6 State of the art

The idea of using analyses of one type to improve analyses of other types is not new. There are some approaches employing LFG to improve dependency parsing or dependency parsing to improve LFG parsing. In the approach by Øvrelid *et al.* (2009), the output of a grammar-driven LFG parser is encoded as additional features in the data-driven dependency parsing models. Çetinoğlu *et al.* (2010) train a dependency parser on LFG-inspired dependency trees generated either with 'LFG constituency parsing pipeline' or 'LFG dependency parsing pipeline'. Çetinoğlu *et al.* (2013) in turn propose a dependency-based sentence simplification approach. The simplification consists in deleting erroneous parts from unparsed sentences (the erroneous parts are identified on the basis of dependency structures of considered sentences). Sentences that fail to have a complete analysis in their original form are simplified this way and parsed with XLE, in the hope of receiving a coherent – even if incomplete – analysis.

Furthermore, Sagae *et al.* (2007) use a dependency parser to restrict the search space of a more complex HPSG parser. Output of a statistical dependency parser serves as constraints (hard or soft) to improve the HPSG parsing. The HPSG parser produces parse trees that strictly conform to the output of the dependency parser (hard dependency constrains). Some dependency structures do not conform to HPSG schema used in parsing. Predetermined dependencies are therefore treated as soft constraints that do not prohibit schema applications but penalise the log-likelihood of partial parse trees created by schema application that violate the dependency constraints.

To the best of our knowledge, using clues from a simple dependency parser to recompose deeply-parsed fragments of a sentence not completely analysable by a deep parser, is a novel contribution of this work.

7 Future work

While the results reported above are quite promising, there is still room for improvement. First, not all FRAGMENT analyses could be converted into proper f-structures. Some of them contain strings of tokens that could not be analysed as correct phrases by POLFIE rules and are annotated as TOKENs. In the worst case the entire sentence is annotated as a TOKEN. FRAGMENT f-structures with TOKENs are currently not modified, but as they might correspond to proper sentences, it would be useful to develop a procedure of modifying them. Second, it should be verified whether it is possible to disambiguate multiple solutions which are output for a sentence based on a dependency tree of this sentence. Finally, another possibility to investigate is to provide a dependency parser with a sentence partially parsed by XLE. Then, the initial configuration of a transition-based parser could correspond to the set of relations in a FRAGMENT f-structure,[4] or – in the graph-based approach – arcs of a directed graph corresponding to relations of a FRAGMENT f-structure could be initially scored high so that they are selected to build the final dependency tree.

Acknowledgements

We are grateful to the anonymous reviewers for comments which led to multiple improvements in the form and contents of this paper. Work reported here has been partially financed by the Polish Ministry of Science and Higher Education within the CLARIN ERIC programme 2015–2016 (`http://clarin.eu/`).

References

Bohnet, B. (2010). Very High Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING 2010, pages 89–97.

Bresnan, J., editor (1982). *The Mental Representation of Grammatical Relations*. MIT Press Series on Cognitive Theory and Mental Representation. The MIT Press, Cambridge, MA.

Bresnan, J. (2001). *Lexical-Functional Syntax*. Blackwell Textbooks in Linguistics. Blackwell, Malden, MA.

Çetinoğlu, O., Foster, J., Nivre, J., Hogan, D., Cahill, A., and van Genabith, J. (2010). LFG Without C-Structures. In *Proceedings of the 9th International Workshop on Treebanks and Linguistic Theories*, pages 43–54.

Çetinoğlu, O., Zarrieß, S., and Kuhn, J. (2013). Dependency-based sentence simplification for increasing deep LFG parsing coverage. In *Proceedings of the LFG13 Conference*, pages 191–211.

[4] As suggested in one of the reviews.

Crouch, D., Dalrymple, M., Kaplan, R., King, T., Maxwell, J., and Newman, P. (2011). *XLE Documentation*. Palo Alto Research Center (PARC), Palo Alto, CA.

Dalrymple, M. (2001). *Lexical Functional Grammar*. Academic Press, San Diego, CA.

Maxwell III, J. T. and Kaplan, R. M. (1993). The Interface between Phrasal and Functional Constraints. *Computational Linguistics*, **19**(4), 571–590.

Øvrelid, L., Kuhn, J., and Spreyer, K. (2009). Improving Data-Driven Dependency Parsing Using Large-Scale LFG Grammars. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (Conference Short Papers)*, pages 37–40.

Patejuk, A. and Przepiórkowski, A. (2012). Towards an LFG parser for Polish: An exercise in parasitic grammar development. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012*, pages 3849–3852, Istanbul, Turkey. ELRA.

Patejuk, A. and Przepiórkowski, A. (2014). Synergistic development of grammatical resources: A valence dictionary, an LFG grammar, and an LFG structure bank for Polish. In V. Henrich, E. Hinrichs, D. de Kok, P. Osenova, and A. Przepiórkowski, editors, *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT 13)*, pages 113–126, Tübingen, Germany. Department of Linguistics (SfS), University of Tübingen.

Pollard, C. and Sag, I. A. (1987). *Information-Based Syntax and Semantics, Volume 1: Fundamentals*. Number 13 in CSLI Lecture Notes. CSLI Publications, Stanford, CA.

Pollard, C. and Sag, I. A. (1994). *Head-driven Phrase Structure Grammar*. Chicago University Press / CSLI Publications, Chicago, IL.

Przepiórkowski, A., Bańko, M., Górski, R. L., and Lewandowska-Tomaszczyk, B., editors (2012). *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, Warsaw.

Riezler, S., King, T. H., Kaplan, R. M., Crouch, R., Maxwell, J. T., and Johnson, M. (2002). Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 271–278.

Sagae, K., Miyao, Y., and Tsujii, J. (2007). HPSG Parsing with Shallow Dependency Constraints. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 624–631.

Wróblewska, A. (2014). *Polish Dependency Parser Trained on an Automatically Induced Dependency Bank*. Ph. D. dissertation, Institute of Computer Science, Polish Academy of Sciences, Warsaw.

A Survey of Multiword Expressions in Treebanks

Victoria Rosén,[1] Gyri Smørdal Losnegaard,[1]
Koenraad De Smedt,[1] Eduard Bejček,[2] Agata Savary,[3]
Adam Przepiórkowski,[4,5] Petya Osenova[6,7] and Verginica Barbu Mititelu[8]

[1]University of Bergen, [2]Charles University in Prague,
[3]François Rabelais University of Tours,
[4]Institute of Computer Science, Polish Academy of Sciences,
[5]University of Warsaw, [6]Sofia University St. Kl. Ohridski,
[7]Institute of Information and Communication Technologies,
Bulgarian Academy of Sciences,
[8]Romanian Academy Research Institute
for Artificial Intelligence "Mihai Drăgănescu"

E-mail: {victoria|gyri.losnegaard|desmedt}@uib.no,
bejcek@ufal.mff.cuni.cz, agata.savary@univ-tours.fr,
adamp@ipipan.waw.pl, petya@bultreebank.org, vergi@racai.ro

Abstract

We present the methodology and results of a survey on the annotation of multiword expressions in treebanks. The survey was conducted using a wiki-like website filled out by people knowledgeable about various treebanks. The survey results were studied with a comparative focus on prepositional MWEs, verb-particle constructions and multiword named entities.

1 Introduction

There is currently little agreement on how multiword expressions (MWEs) should be annotated in treebanks, and there is, in fact, not even agreement on what constitutes a MWE in NLP. This makes it difficult to study and exploit MWEs in language resources, including treebanks.

PARSEME[1] is a COST Action dedicated to the study of MWEs. PARSEME's working group 4 is concerned with the annotation of MWEs in treebanks. One of the intended outcomes of this working group is to make recommendations for common principles and guidelines for annotating MWEs in treebanks. As a step towards making such recommendations, we have made a survey of the ways in which

[1] http://www.parseme.eu/

different types of MWEs are currently annotated in a variety of treebanks. This survey was performed by asking people knowledgeable about particular treebanks to describe the annotation of different types of MWEs by filling out an online form. It has not been the goal of the present study to check to what extent the principles and guidelines for each treebank have been followed.

The paper is structured as follows: In section 2 the methodology of gathering and summarizing data is presented. Section 3 presents a summary of preliminary findings for three MWE types. Section 4 concludes the paper.

2 Methodology

A structured survey form was set up by establishing a wiki with editable pages written in a Wikimedia-like framework and featuring a simple markup language and easy hyperlinking. The main page of the wiki contains a table which we will call the 'survey table' and which is shown in Figure 1. The main page also presents detailed instructions for entering information.

There is a row in the survey table for each treebank for which information has been collected. The row name (in the first column of the table) is the name of the treebank. The second column contains the language, and the third the annotation type of the treebank. The remaining columns are for MWE types. All cells with blue in the survey table are clickable and lead to embedded information pages.[2] The next sections present the elements in the table in more detail.

2.1 The treebanks

The survey is open-ended and will continue to be updated with information about different treebanks until the end of the PARSEME action in the spring of 2017. Currently, information has been gathered about 17 treebanks for 15 languages. The two main types are dependency and constituency treebanks.

The dependency treebanks are (the language is shown in parentheses when it is not included in the name of the treebank):

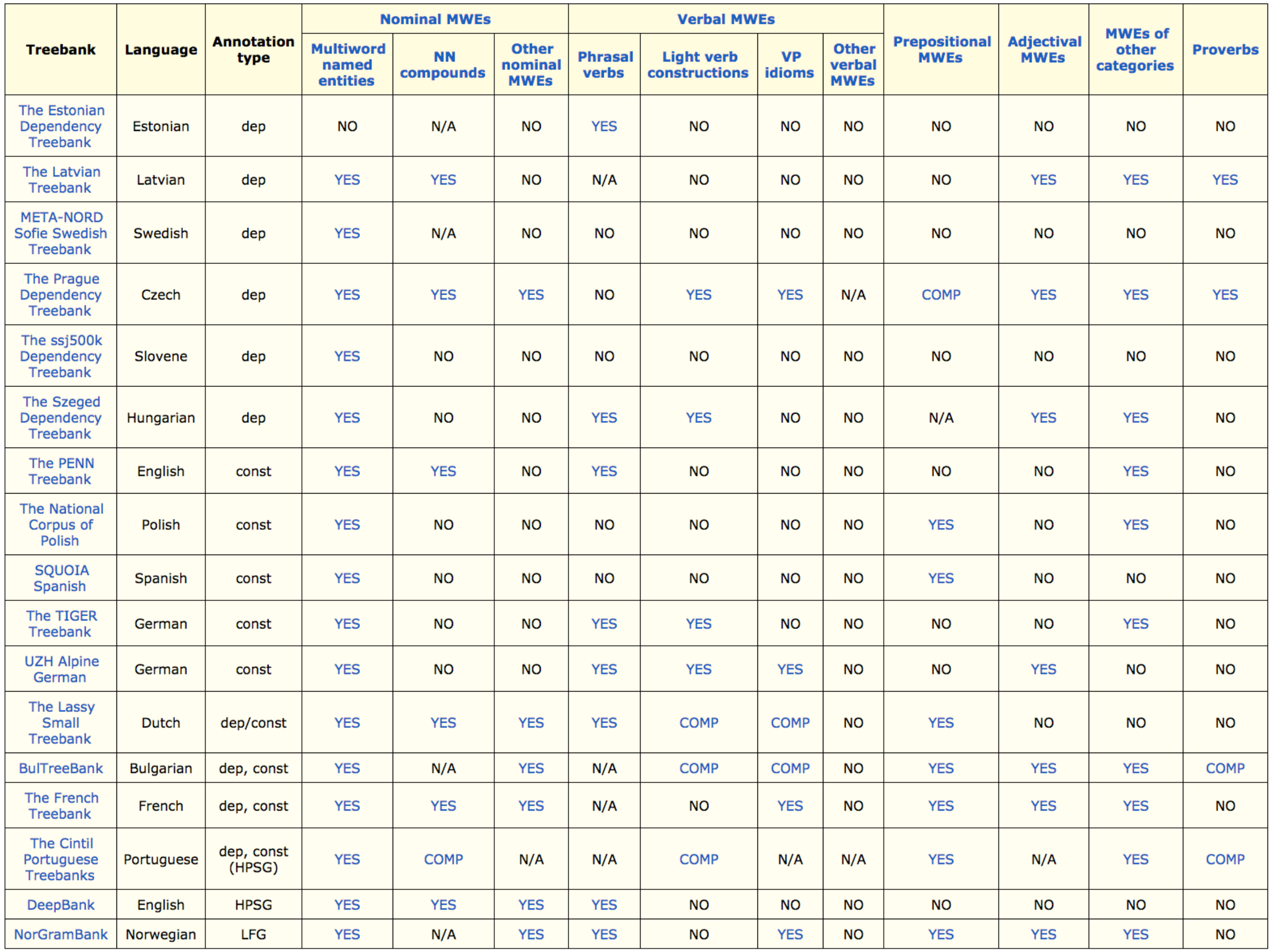
- The Estonian Dependency Treebank [11]
- The Latvian Treebank [13]
- The META-NORD Sofie Swedish Treebank [10]
- The Prague Dependency Treebank (Czech) [3]
- The ssj500k Dependency Treebank (Slovene) [7]
- The Szeged Dependency Treebank (Hungarian) [18]

The constituency treebanks include:

- The National Corpus of Polish [9, 15]
- The PENN Treebank (English)[3]

[2] For the online version, see http://clarino.uib.no/iness/page?page-id=MWEs_in_Parseme

[3] http://www.cis.upenn.edu/~treebank/



Treebank	Language	Annotation type	Nominal MWEs			Verbal MWEs				Prepositional MWEs	Adjectival MWEs	MWEs of other categories	Proverbs
			Multiword named entities	NN compounds	Other nominal MWEs	Phrasal verbs	Light verb constructions	VP idioms	Other verbal MWEs				
The Estonian Dependency Treebank	Estonian	dep	NO	N/A	NO	YES	NO	NO	NO	NO	NO	NO	NO
The Latvian Treebank	Latvian	dep	YES	YES	NO	N/A	NO	NO	NO	NO	YES	YES	YES
META-NORD Sofie Swedish Treebank	Swedish	dep	YES	N/A	NO	NO	NO	NO	NO	NO	NO	NO	NO
The Prague Dependency Treebank	Czech	dep	YES	YES	YES	NO	YES	YES	N/A	COMP	YES	YES	YES
The ssj500k Dependency Treebank	Slovene	dep	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
The Szeged Dependency Treebank	Hungarian	dep	YES	NO	NO	YES	YES	NO	NO	N/A	YES	YES	NO
The PENN Treebank	English	const	YES	YES	NO	YES	NO	NO	NO	NO	NO	YES	NO
The National Corpus of Polish	Polish	const	YES	NO	NO	NO	NO	NO	NO	YES	NO	YES	NO
SQUOIA Spanish	Spanish	const	YES	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO
The TIGER Treebank	German	const	YES	NO	NO	YES	YES	NO	NO	NO	NO	YES	NO
UZH Alpine German	German	const	YES	NO	NO	YES	YES	YES	NO	NO	YES	NO	NO
The Lassy Small Treebank	Dutch	dep/const	YES	YES	YES	YES	COMP	COMP	NO	YES	NO	NO	NO
BulTreeBank	Bulgarian	dep, const	YES	N/A	YES	N/A	COMP	COMP	NO	YES	YES	YES	COMP
The French Treebank	French	dep, const	YES	YES	YES	N/A	NO	YES	NO	YES	YES	YES	NO
The Cintil Portuguese Treebanks	Portuguese	dep, const (HPSG)	YES	COMP	N/A	N/A	COMP	N/A	N/A	YES	N/A	YES	COMP
DeepBank	English	HPSG	YES	YES	YES	YES	NO	NO	NO	NO	NO	NO	NO
NorGramBank	Norwegian	LFG	YES	N/A	YES	YES	NO	YES	NO	YES	YES	YES	NO

Figure 1: The survey table

- The SQUOIA Spanish Treebank[4]
- The TIGER Treebank (German) [5]
- The UZH Alpine German Treebank[5]

There are six treebanks which cannot be classified simply as either dependency or constituency treebanks. These are:

- BulTreeBank (Bulgarian) [16]
- The French Treebank [1]
- The Lassy Small Treebank (Dutch) [17]
- The CINTIL Treebanks (Portuguese) [4]
- DeepBank (English) [8]
- NorGramBank (Norwegian)[6]

BulTreeBank and the French Treebank offer both constituency and dependency analyses. The Lassy Small Treebank has analyses that are a cross between constituency and dependency graphs. The CINTIL Treebanks and DeepBank are both based on Head Driven Phrase Structure Grammar (HPSG) [12], while NorGramBank is based on Lexical Functional Grammar (LFG) [6].

Clicking on the treebank name (in the first column of the table) brings up a 'treebank description page'. Here information is given such as name, author, formalism, license, links to documentation, history (how the treebank was constructed), whether it is static or dynamic, etc.

2.2 The MWE types

The table headers show the types of MWEs described:

- Nominal MWEs
  - Multiword named entities
  - NN compounds
  - Other nominal MWEs
- Verbal MWEs
  - Phrasal verbs
  - Light verb constructions
  - VP idioms
  - Other verbal MWEs
- Prepositional MWEs
- Adjectival MWEs
- MWEs of other categories
- Proverbs

This typology was based on a discussion of more or less accepted types described in the literature [2, 14], taking into account the trade-off between offering major types as a guidance and allowing other types and subtypes that are found in

[4] http://www.cl.uzh.ch/research/maschinelleuebersetzung/hybridmt_en.html

[5] http://www.cl.uzh.ch/research/parallelcorpora/paralleltreebanks/smultron_en.html

[6] http://clarino.uib.no/iness/

some treebanks. Clicking on a column header for a MWE type opens up a 'MWE type description page'.

2.3 MWE information cells and MWE description pages

Each cell in a MWE type column has one of the following values:

- N/A (for 'not applicable'): the MWE type does not occur in the language
- NO: the MWE type occurs in the language but the treebank lacks annotation for it
- YES: the MWE type is annotated in the treebank
- COMP: the MWE type is not annotated as such, but is analyzed compositionally

Clicking on the value YES or COMP brings up a 'MWE example page' with a detailed description of one or more examples of the MWE type in a particular treebank. Each MWE example page contains the following information (for each example):

- The type of MWE and the treebank name
- An example sentence containing the MWE, with interlinear glosses and an idiomatic translation
- A graphic (screenshot or similar) with a visualization of the analysis
- A prose explanation of the analysis
- A search expression for the MWE and a prose description of what the expression does

By way of illustration, the MWE example page for prepositional MWEs in NorGramBank is given in Figure 2.

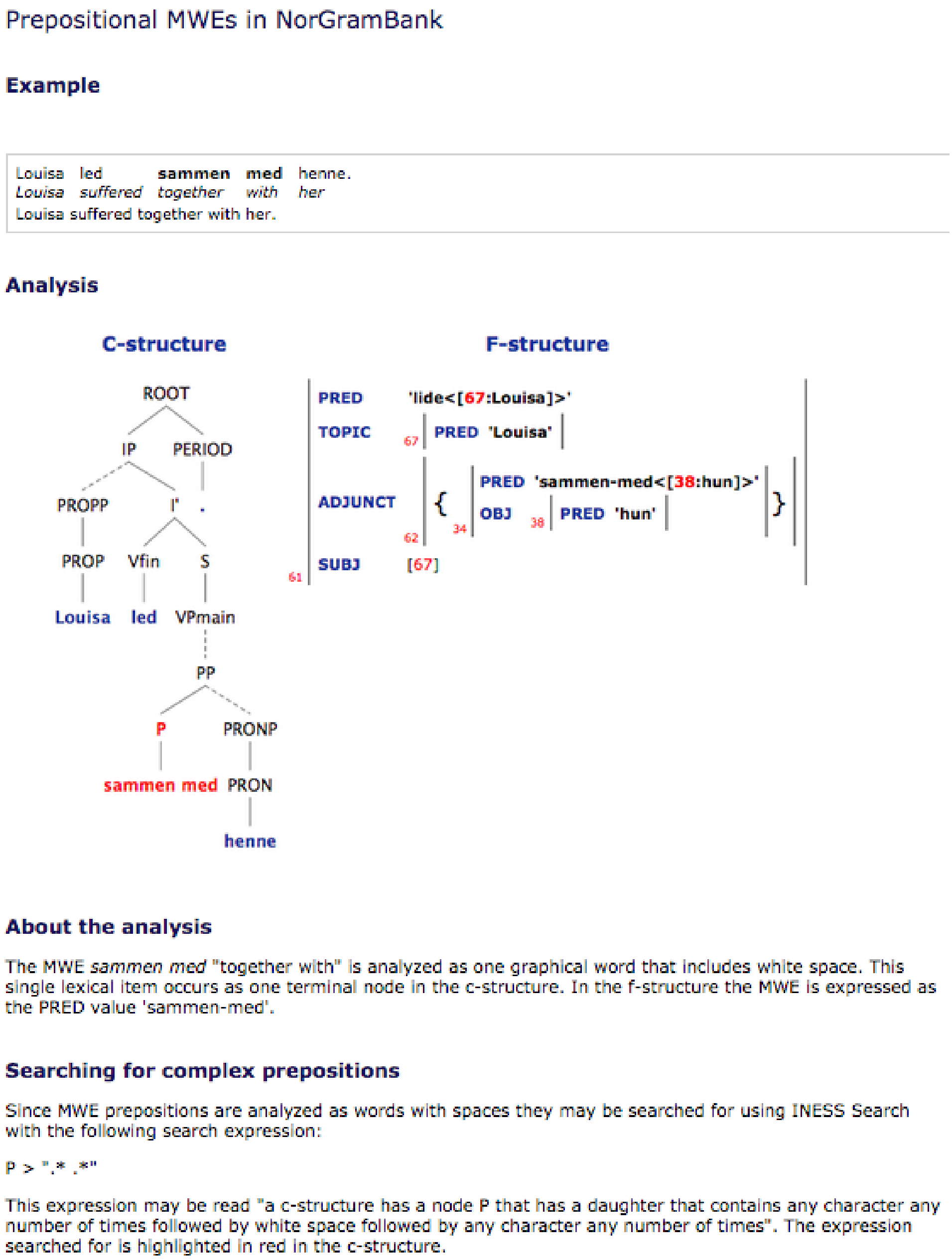
3 Results and discussion

The survey allows comparison of many different types of MWEs along several dimensions. Within the confines of the present paper, we will focus on comparisons for three of the most commonly annotated types of MWEs. Table 1 shows the number of MWEs of various types that are annotated in the survey.

3.1 Prepositional MWEs

Prepositional MWEs are often fixed expressions in Sag et al.'s terminology. Since fixed expressions are lexicalized and do not undergo morphosyntactic variation or internal modification, they can be handled with a words-with-spaces approach [14, p. 192].

Prepositional MWEs are annotated in somewhat different ways in the treebanks in our survey, as illustrated in Figure 3. BulTreeBank and NorGramBank treat them literally as words with spaces, in other words as single graphical words that include white space. The Bulgarian MWE Благодарение на "thanks to" is a terminal node in the tree dominated by *Prep*, while the Norwegian *sammen med* "together



Prepositional MWEs in NorGramBank

Example

Louisa	led	**sammen**	**med**	henne.
Louisa	*suffered*	*together*	*with*	*her*

Louisa suffered together with her.

Analysis

About the analysis

The MWE *sammen med* "together with" is analyzed as one graphical word that includes white space. This single lexical item occurs as one terminal node in the c-structure. In the f-structure the MWE is expressed as the PRED value 'sammen-med'.

Searching for complex prepositions

Since MWE prepositions are analyzed as words with spaces they may be searched for using INESS Search with the following search expression:

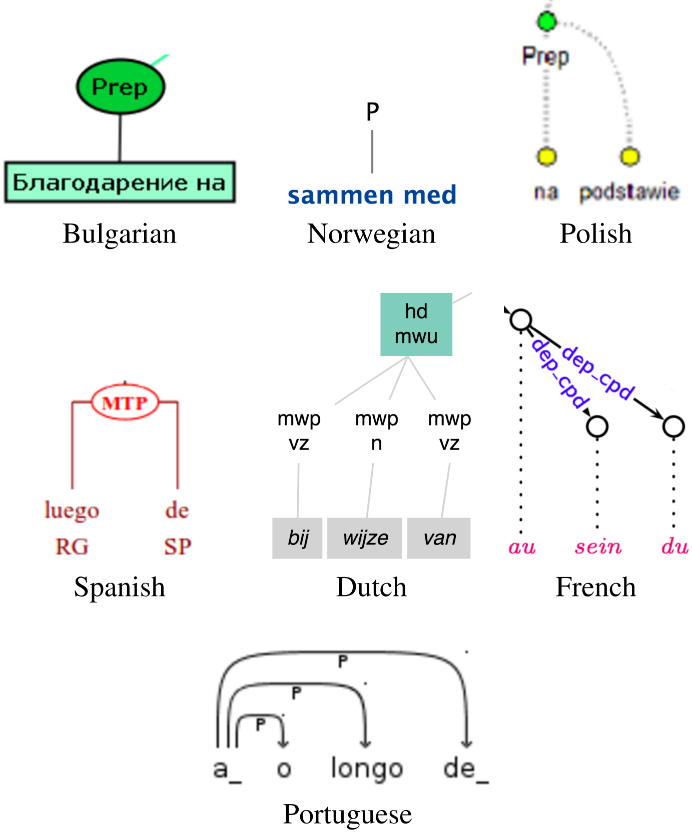
```
P > ".* .*"
```

This expression may be read "a c-structure has a node P that has a daughter that contains any character any number of times followed by white space followed by any character any number of times". The expression searched for is highlighted in red in the c-structure.

Figure 2: MWE example page for prepositional MWEs in NorGramBank

Nominal MWEs	Multiword named entities	16
	NN compounds	6 (+1)
	Others	6
Verbal MWEs	Phrasal verbs	8
	Light verb constructions	4 (+3)
	VP idioms	4 (+2)
	Others	0
Prepositional MWEs		7 (+1)
Adjectival MWEs		7
MWEs of other categories		10
Proverbs		2 (+2)

Table 1: Number of treebanks (out of all 17 treebanks in the survey) with annotations for the different MWE types, with the number of compositional analyses given in parentheses



Bulgarian Norwegian Polish

Spanish Dutch French

Portuguese

Figure 3: Overview of the annotations of prepositional MWEs in seven treebanks

with" is a terminal dominated by *P*. The National Corpus of Polish has a multi-layer annotation, not all of which is shown in the example. Parts of speech are assigned to individual components of a MWE preposition in the morphosyntactic annotation layer (*na* is a preposition and *podstawie* is a noun), and these components are joined into one unit (of type *Prep*) in the syntactic word layer. The SQUOIA Spanish treebank provides a phrasal analysis of the MWE *luego de* "after of", using a special node label *MTP*, and including the PoS labels for the constituents. The LASSY Small Treebank provides a similar analysis of the Dutch MWE *bij wijze van* "by way of", with *mwu* for the mother node and *mwp* for the daughter nodes in addition to the PoS labels for the constituents. The French Treebank provides a left-headed dependency analysis of the MWE *au sein du* "within" (literally "in-the breast of-the"), with *au* as the head and *sein* and *du* as dependents. The Cintil Portuguese Treebanks provide both constituency and dependency analyses; here we show the dependency analysis, which is similar to the one in the French treebank. The MWE *ao longo de* "along" is a left-headed dependency with *ao* as the head. As in French, there are contractions between prepositions and articles, so that the preposition *a* and the article *o* contract to the form *ao*.

Only two of the treebanks treat prepositional MWEs as words with spaces. The other treebanks that annotate these MWEs have separate nodes for their component words, and some of them include part of speech information for these component words. All of these treebanks treat them as prepositions on a syntactic level.

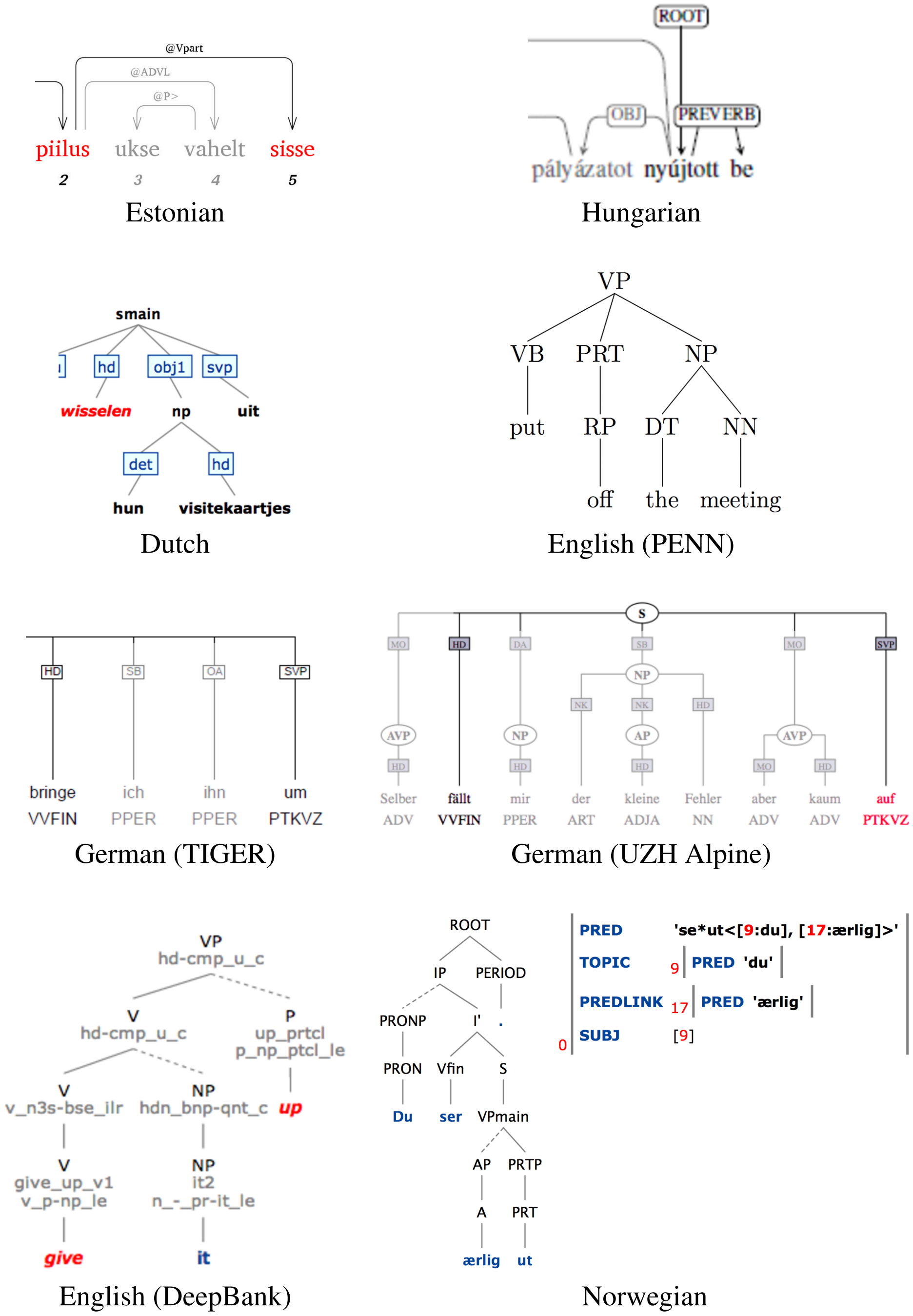
3.2 Verb-particle constructions

Sag et al. consider verb-particle constructions to be an important type of syntactically flexible expressions. These constructions cannot be treated as words with spaces since other words may intervene between the verb and the particle. They cannot simply be treated as compositional either, among other things because the particles often "assume semantics idiosyncratic to verb-particle constructions" [14, p. 194].

In the survey table there is one column for phrasal verbs. Clicking on the column header brings up a page with descriptions of the types of MWE annotations that should be entered in this column:

- Particle verbs such as *show up*
- Verbs with selected prepositions such as *think of*
- Verbs with both particles and selected prepositions such as *come up with*

Some of the languages in the survey do not have phrasal verbs of these three types; Bulgarian, Czech, French, Latvian and Portuguese have *N/A* for "not applicable" in the phrasal verbs column. Swedish, Slovene, Polish and Spanish have *NO* in this column, meaning that the language has the construction but that the treebank lacks annotation for it. Particle verbs are annotated in eight of the treebanks in various ways which reflect their MWE status. Figure 4 includes screenshots of the relevant parts of the analyses for these eight treebanks.



Estonian

Hungarian

Dutch

English (PENN)

German (TIGER)

German (UZH Alpine)

English (DeepBank)

Norwegian

Figure 4: Overview of the annotations of particle verbs in eight treebanks

The annotations for particle verbs are quite similar across the treebanks that have them. In the Estonian treebank there is a VPART dependency from the verb to the particle. The Hungarian treebank has a PREVERB dependency from the verb to the particle; sometimes this particle is a prefix on the verb, and sometimes, as here, it is a separate graphical word. Dutch marks the verb particle as SVP for "separable verb prefix", since, as in Hungarian, it can sometimes form one word with the verb and sometimes, as in the example, occur as a separate word. In the three constituency treebanks with verb-particle constructions, the particle is annotated as a separate constituent in the S or VP that dominates it. The PENN treebank uses the PoS tag RP dominated by PRT; both of the German treebanks use the PoS tag PTKVZ for Partikel Verbzusatz dominated by an SVP node. DeepBank and NorGramBank do not only annotate the particle as a separate constituent, but also incorporate it into the verb in different ways. The DeepBank preterminal of the verb indicates that the verb *give* in this case has a lexical entry which specifies the complement *up*. In NorGramBank, the particle PRT is dominated by a particle phrase PRTP in the c(onstituent)-structure, but it does not contribute any predicate (PRED) of its own to the f(unctional)-structure. The particle is, however, integrated into the PRED for the verb, which is *se*ut*, meaning "look". These latter two annotations make more explicit that the predicate cannot simply be analyzed compositionally.

The annotations for particle verbs turn out to be surprisingly similar across treebanks. The challenge in annotating these constructions is not in how they should be annotated, but in finding the verb-particle constructions themselves.

3.3 Multiword named entities

Of sixteen treebanks for which information is provided for multiword named entities, twelve have examples of person names. In spite of the fact that person names themselves are very similar across the languages in the survey, we do see differences in their annotation. As an illustration, three examples from dependency treebanks are given in Figure 5. In Czech and Swedish there is a dependency between the first and last names, but in Czech the last name is the head, whereas in Swedish the first name is the head. In Latvian, there is a special node called 'namedEnt' which has both the first and the last names as dependents.

In addition to person names, there are several other types of multiword named entities which are exemplified: geographical names, names of institutions and organizations, temporal expressions such as dates and times, etc. Nine types of multiword named entities are distinguished in the Prague Dependency Treebank: person, institution, location, object, address, biblio, time, foreign and number. The National Corpus of Polish has six main types (persName, orgName, geogName, placeName, date and time), and there are eight subtypes. For most treebanks in the survey, however, only one or two examples are given, without it being clear if other types are also annotated.

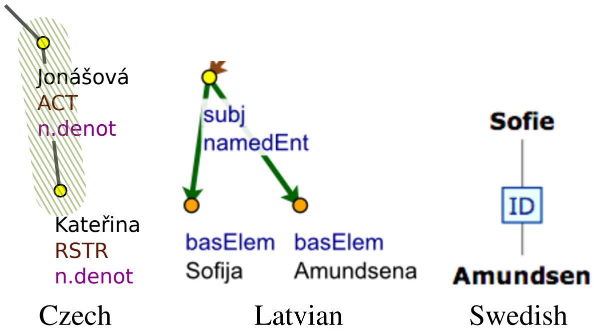
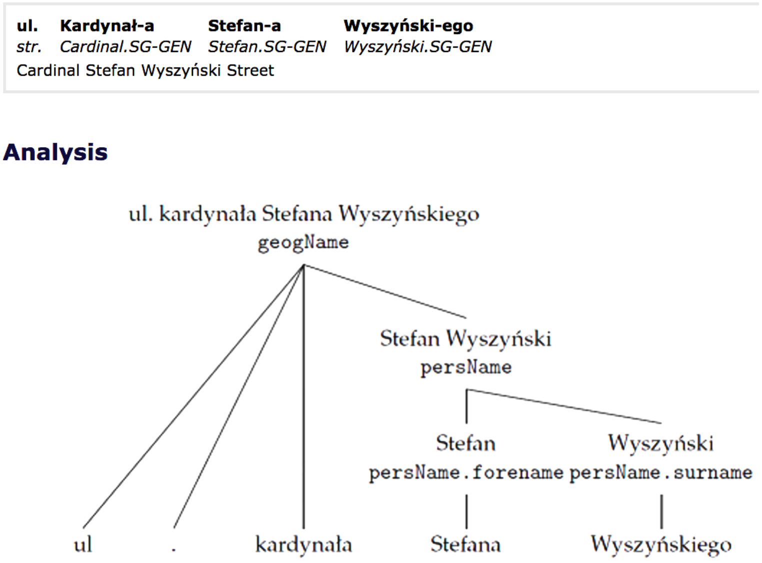


Figure 5: Examples of MWE person names in three dependency treebanks

An example of a geographical named entity from the National Corpus of Polish is given in Figure 6. This is a complex example where the annotation of a person name is embedded inside the annotation of a geographical name. We note, however, that *Kardynała* 'Cardinal' is annotated as part of the geographical name, whereas it is actually a title that belongs hierarchically to a different level in the analysis. How such titles should be treated is an important question in itself. In the Dutch treebank, the title *drs.* is considered part of the named entity, as shown in Figure 7.



ul.	**Kardynał-a**	**Stefan-a**	**Wyszyński-ego**
str.	*Cardinal.SG-GEN*	*Stefan.SG-GEN*	*Wyszyński.SG-GEN*

Cardinal Stefan Wyszyński Street

Analysis

Figure 6: Example of a MWE named entity annotation in the Polish National Corpus

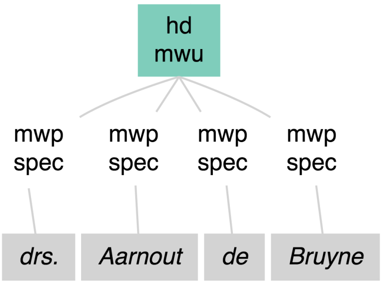


Figure 7: Example of a MWE named entity annotation in Lassy

A complex example from the ssj500k Dependency Treebank for Slovene is the analysis of the organization name *Odbor Združenih narodov za odpravo diskriminacije žensk* “The United Nations Committee on the Elimination of Discrimination against Women”. In this treebank, multiword named entities are annotated as chunks of connected tokens on the morphosyntactic layer. The whole entity is also labeled as a proper/organization name (*stvarno*). The dependencies are shown in Figure 8.

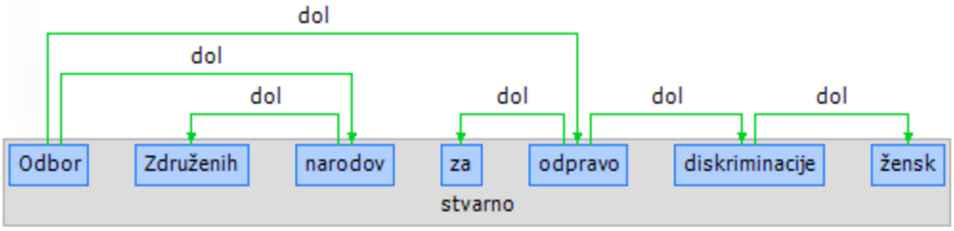


Figure 8: Example of a MWE named entity annotation in the ssj500K Dependency Treebank for Slovene

In conclusion, the annotation of multiword names ranges from very simple structures, similar to fixed expressions, to more complex structures, sometimes with other names embedded inside them. Treebanks may also vary considerably as to the types of named entities that they distinguish. This initial study shows that the survey should request more information about the range of possible annotations for multiword named entities in each treebank.

4 Conclusion and future work

We have reported on the first results from a focused survey on MWEs in various treebanks. We have developed a simple MWE typology, taking seminal works as a starting point. The survey includes treebanks with different annotation types.

While some MWEs are language specific (e.g. verb-particle constructions that are typical for Germanic languages), others occur in all the languages for which we have information (e.g. named entities).

The results indicate that for some MWE types (e.g. multiword named entities) there is more variation in annotation approaches than for other types (e.g. prepositional MWEs and verb-particle constructions).

Our study has also shown that better treebank documentation is important. It is often difficult to interpret the examples if there is no clear link to the tagset, the annotation guidelines, and similar information.

The survey is open-ended and can accommodate entries for additional languages and treebanks. The results of the survey are a step towards making recommendations for common principles and guidelines for annotating MWEs in treebanks.

Acknowledgments

This work has been supported in part by the PARSEME European Cost Action (IC1207) and the COST CZ grant of the MEYS of the Czech Republic (LD14117).

References

[1] Anne Abeillé, Lionel Clément, and François Toussenel. Building a treebank for French. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, volume 20 of *Text, speech and language technology*. Kluwer Academic Publishers, Dordrecht, 2003.

[2] Timothy Baldwin and Su Nam Kim. Multiword expressions. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing*, chapter 12. CRC Press, Boca Raton, FL, USA, 2nd edition, 2010.

[3] Eduard Bejček, Eva Hajičová, Jan Hajič, Pavlína Jínová, Václava Kettnerová, Veronika Kolářová, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Jarmila Panevová, Lucie Poláková, Magda Ševčíková, Jan Štěpánek, and Šárka Zikánová. Prague Dependency Treebank 3.0, 2013. Data, `http://hdl.handle.net/11858/00-097C-0000-0023-1AAF-3`.

[4] António Branco, Francisco Costa, João Silva, Sara Silveira, Sérgio Castro, Mariana Avelãs, Clara Pinto, and João Graça. Developing a deep linguistic databank supporting a collection of treebanks: the CINTIL DeepGramBank. In *LREC*, 2010.

[5] Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. TIGER: Linguistic interpretation of a German corpus. *Research on Language and Computation*, 2(4):597–620, 2004.

[6] Mary Dalrymple. *Lexical Functional Grammar*, volume 34 of *Syntax and Semantics*. Academic Press, San Diego, CA, 2001.

[7] Tomaz Erjavec, Darja Fiser, Simon Krek, and Nina Ledinek. The JOS linguistically tagged corpus of Slovene. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, page 1806–1809, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[8] Dan Flickinger, Yi Zhang, and Valia Kordoni. Deepbank: A dynamically annotated treebank of the Wall Street Journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, pages 85–96, 2012.

[9] Katarzyna Głowińska and Adam Przepiórkowski. The Design of Syntactic Annotation Levels in the National Corpus of Polish. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[10] Gyri Smørdal Losnegaard, Gunn Inger Lyse, Anje Müller Gjesdal, Koenraad De Smedt, Paul Meurer, and Victoria Rosén. Linking Northern European infrastructures for improving the accessibility and documentation of complex resources. In Koenraad De Smedt, Lars Borin, Krister Lindén, Bente Maegaard, Eiríkur Rögnvaldsson, and Kadri Vider, editors, *Proceedings of the workshop on Nordic language research infrastructure at NODALIDA 2013, May 22–24, 2013, Oslo, Norway. NEALT Proceedings Series 20*, number 89 in Linköping Electronic Conference Proceedings, pages 44–59. Linköping University Electronic Press, 2013.

[11] Kadri Muischnek, Kaili Müürisep, Tiina Puolakainen, Eleri Aedmaa, Riin Kirt, and Dage Särg. Estonian dependency treebank and its annotation scheme. In *Proceedings of 13th Workshop on Treebanks and Linguistic Theories (TLT13)*, pages 285–291, 2014.

[12] Carl Pollard and Ivan A Sag. *Head-driven phrase structure grammar*. University of Chicago Press, 1994.

[13] Lauma Pretkalnina and Laura Rituma. Syntactic issues identified developing the Latvian treebank. In *Baltic HLT*, pages 185–192, 2012.

[14] Ivan Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *Lecture*

Notes in Computer Science. Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, volume 2276, pages 189–206. Springer, 2002.

[15] Agata Savary, Jakub Waszczuk, and Adam Przepiórkowski. Towards the Annotation of Named Entities in the Polish National Corpus. In *Proceedings of LREC 10, Valletta, Malta*. European Language Resources Association, 17-23 May 2010.

[16] Kiril Simov, Petya Osenova, Alexander Simov, and Milen Kouylekov. Design and implementation of the Bulgarian HPSG-based treebank. *Journal of Research on Language and Computation*, Special Issue:495–522, 2005.

[17] Gertjan van Noord. Huge parsed corpora in LASSY. In Frank Van Eynde, Anette Frank, Koenraad De Smedt, and Gertjan van Noord, editors, *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT7)*, pages 115–126. LOT, 2009.

[18] Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. Hungarian dependency treebank. In *LREC*, 2010.

Morphological Triggers of Syntactic Changes: Treebank-Based Information Theoretic Approach*

Alexandra Simonenko, Benoit Crabbé and Sophie Prévost

ALPAGE, LaTTiCe
E-mail: alexandra.simonenko@mail.mcgill.ca,
benoit.crabbe@linguist.univ-paris-diderot.fr,
sophie.prevost@ens.fr

Abstract

This paper addresses the classic problem of the triggers of the passage from a relatively free word order to a strict SVO in the history of French ([26], [9], [8], [28], [12], [7]). We present a corpus-based modelling of two, likewise classic, lines of analysis. First, we explore the link between the loss of word order freedom and the disappearance of morphological case marking ([22], [23], [6], [20], [14]). Second, we evaluate the syncretisation of verbal agreement and massive appearance of overt preverbal subject pronouns ([1], [19], [21]) as a potential analogical trigger of a generalized SVO (e.g. [3] for an analogy-based explanation of the change in nominal syntax in Old English). Although the analytical intuitions themselves have a long history, only recently has it become possible to perform their quantitative evaluations due to the availability of large (for historical data) annotated treebanks of Medieval French ([15], [16], and [24]).

1 Introduction

This paper presents a quantitative corpus-based investigation of the possible causes of the fixation of the word order in the history of French using Information Theoretic measures. Medieval French (MF) went from a (loose) V2 system, permitting for all six permutations of S, O, and V, to a relatively strict SVO (e.g. [8], [12]).

(1) [L' altre meitet]$_{obj}$ avrat$_{v}$ Rollant$_{sbj}$
the other half will.have Roland
'Roland will have the other half' (1100-ROLAND-V,36.446) V2 in Old French

*We gratefully acknowledge funding support from Labex EFL, grant ANR-10-LABX-0083. We are also thankful to three anonymous reviewers of TLT 2015 for very helpful feedback.

(2) Roland$_{sbj}$ aura$_{v}$ l' autre moitié$_{obj}$
Roland will.have the other half
'Roland will have the other half' SVO in Modern French

(3) *L' autre moitié$_{obj}$ aura$_{v}$ Roland
the other half will.have Roland
*V2 in Modern French

The passage to SVO has been frequently attributed to the disappearance of morphological case marking, on the assumption that linear position and case both can mark syntactic roles and therefore the former can substitute for the latter (e.g. [27, 289], [6]). This is at the least a plausible analysis for French since by the X century, the distinction between nominative and accusative in MF was mostly retained only for masculine nouns (e.g. *reis*$_{nom,sg}$, *rei*$_{acc,sg}$), and even that was becoming unstable ([12]), as illustrated by the unmarked subject in (5).

(4) **Reis Chielperics** tam bien en fist...
king Chilpéric so well of.it made
'King Chilperic dealt with it so well...' (0980-LEGER-V,XII.80)

(5) É li nostre **rei** nus jugerá...
and the our king us will.judge
'And our king will judge us.' (1150-QUATRELIVRE-P,17.529)

In the typological perspective, the existence of some sort of an inverse dependency between the fixedness of the word order (i.e. arguments having strict positions with respect to the predicate: either SVO or OVS) and the availability of morphological case marking has been claimed to hold in the literature ranging from [22] to [2]. However, the position-for-case substitution in MF has remained in the hypothetical realm since until recently it had been virtually impossible to quantify the relevant changes. Another difficulty consisted in the absence of comparable measures of the contributions of the two markers for the syntactic role identification. Below we propose a way to circumvent both problems by using Information Theoretic notions and distributions drawn from MCVF.

MCVF is a treebank of tagged, parsed and functionally annotated French texts from X to XVIII cc. with Penn treebank style annotation scheme (approx. 1 mln words). We used CorpusSearch, a tool for matching tree patterns in corpora, which can search for the relations of precedence and dominance, for specific morphological forms as well as code utterances for parameters such as word order and presence of an overt subject.[1]

In addition, we explore a second, and compatible, explanation of the passage to SVO. It has at its core the syncretisation of verbal subject agreement suffixes, and the massive emergence of pronominal subjects. The argument runs as follows: syncretisation of verbal agreement led to the replacement of pro-drop by overt pronominal subjects. The latter, being prevailingly preverbal, triggered reanalysis

[1] `http://corpussearch.sourceforge.net/`

of the position of all subjects, including nominal, as preverbal. In order to build a quantitative model, we propose to treat verbal subject agreement as a signal of subject's person feature and to quantify it using, once again, entropy measures. We then compare temporal profiles of agreement syncretisation and pronominal subject expression. Finally, we compare the rate of subject expression and the rate of preverbal nominal subjects to see if the two are correlated.

2 Loss of morphological case and word order flexibility

2.1 Morphology-syntax tradeoff hypothesis

Since both word order and morphological changes manifest themselves as gradual replacements of one alternative by another over the centuries rather than "overnight" categorical shifts, establishing a temporal relation between the two has been virtually impossible until very recently due to the absence of tools for quantifying the relevant changes. Establishing the temporal profiles of the changes is, in turn, indispensable for modelling grammatical relations (if any) between the corresponding phenomena. These points seem to been overlooked in the debate about the relationship between case and word order, which led to claims such as the following one from [10, 22]: "a ... complication with the theory that phonetic attrition of the classical Latin case system necessitated a fixed Romance SVO order is that it is simply not true. ... [L]ate Latin and early Romance retained at least a binary case system (nominative vs. oblique) and were characterized by Verb Second constraint, such that SVO was just one of many possible word orders. From this we can only conclude that there is no necessary causal relation between phonetic attrition, in this case acting upon the case system, and the emergence of analytic structural changes." As we show below, such conclusions are unwarranted by the corpus data, given that the *robustness* of nominative marking, estimated based on the proportion of nominative marked subjects among all subjects, was different at different points in time (overall decreasing), and so was the robustness of linear position marking (overall increasing).[2] The mere fact that in a given text we find both nominative marked subjects and SVO orders does not necessarily speaks for or against a particular relation between case and order. In the following section we propose a way to track diachronic changes in the distribution of case markers and linear orders and to measure their contribution to the identification of syntactic functions.

[2] Note that our approach is very different from approaches evaluating the role of case based on considering all factors, lexical and grammatical (e.g. verbal semantics and discourse context), which could potentially be used as keys for recovering grammatical functions ([23], [17]). While those studies evaluate how often morphological case was crucial for recovering grammatical functions (e.g. [17, 62] estimates that it was the case only in 5-10% of utterances in Late Latin), we are estimating its unambiguity as a signal (see below).

2.2 Methodology

Building on the classic insight of [5] and others that morphological case and linear position can be used to signal syntactic roles, we propose a way to quantify their efficiency using Shannon's entropy in order to give them a common quantificational expression. We start with a working "tradeoff" hypothesis: the expectation that as one signal weakens, an alternative signal gets stronger. Informally, the strength of a signal, its efficiency, is a measure of a marker's unambiguity. To illustrate this, imagine that in one text among arguments with accusative marking there are 80% of direct objects and 20% of subjects, while in another text the proportions are 50% and 50% respectively. Informally, accusative marker is a less ambiguous in the first text than in the second, where it is maximally ambiguous.

This can be formalized using conditional entropy measures. Let X and Y be two discrete random variables, the conditional entropy is the quantity:

$$H[Y|X] = -\sum_{x \in X} P(X=x) \sum_{y \in Y} P(Y=y|X=x) \log_2 P(Y=y|X=x) \qquad (1)$$

where Y is the dependent variable, conditioned on some context X. In our example, Y is a grammatical function, subject or object and X represents the context of the dependent in terms of its position with respects to the head or its case properties[3].

In the next section, we describe a method for estimating the conditional entropies $H[\textsc{function}|\textsc{case}]$ and $H[\textsc{function}|\textsc{position}]$ using distributions from [15] and [16].

2.3 Data extraction

The corpora are morphologically and syntactically annotated using Penn Treebank kind of annotations. It consist of 35 texts from 980 to 1740, which gives about 1 mln words. We extracted all clauses with a finite verb form and a dependent, being either an overt nominal subject or a nominal object. We included only the nouns belonging to the traditional first declension class (e.g. *reis* "king"). As a preliminary step, we manually defined the declension class of each noun form in the corpus and listed them separately. This step was necessary since morphological case marking was not operative in the second declension class (*femme* "woman") during the attested periods and we had to exclude it from our study of the case marking evolution. We also excluded nouns featuring suppletive case marking (e.g. ber_{nom} vs. $baron_{nom}$ 'baron'), as well as nouns whose stems end in *s/z/x*, since for those case marking is neutralised. There is a total of 15,768 examples for subjects and 10,033 examples for objects. Each example is coded with the following variables:

[3] Although our models may look similar to those of [2], one should observe that their goal is opposite: [2] tries to measure to which extent the dependency structure is a good predictor of word order, whereas in our case we try to predict the dependency type given word order and case.

1. DATE. Each clause was coded for the date of the text from which it was taken (e.g. *980*, *1155* etc.): our query matched the identifier node appended to every finite clause with the date attributed to a given text by a scholarly consensus.[4]

2. Syntactic FUNCTION. Every clause was coded as containing an overt nominal subject – *sbj* – or a nominal object – *obj*.[5] Clauses containing subjects are those clauses with a constituent NP-SBJ dominating one of the following four tags: NCS, NCPL, NPRS, NPRPL, which correspond to common singular noun, common plural noun, proper singular noun, and proper plural noun respectively (see Fig. 1). Clauses containing objects are those with a constituent NP-ACC dominating a nominal tag, (Fig. 2).

3. POSITION of the dependant with respect to the finite verb.[6]

   - The code *pre* was assigned if the dependent NP constituent precedes linearly the finite verb tag (AJ, EJ, LJ, MDJ or VJ in [15]).
   - The code *post* was assigned if the dependent NP constituent follows linearly the finite verb tag (AJ, EJ, LJ, MDJ or VJ in [15]).

4. Morphological CASE.

   - The code *nom* was given to forms ending in *s/z/x* in singular and zero in plural (nominative marking)
   - The code *acc* was given to forms that have no ending in singular and *s/z/x* in plural (accusative marking);

Figure 1 is an example of a coded clause with a nominal subject. The clause is taken from *La Chanson de Roland*, a poem dated from around 1100 and containing a preverbal nominal subject in singular and ending with *s* (nominative pattern).

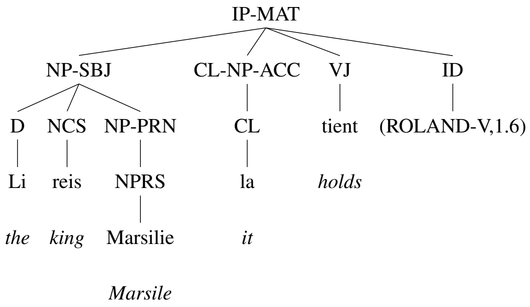
Figure 2 is another example from *La Chanson de Roland*. It illustrates a coded clause with a preverbal nominal object in singular with a zero ending (accusative pattern).

Finally, we use an additional PERIOD factor partitioning our extracted observations by century intervals. For each such PERIOD, we estimated the conditional entropies $H[\text{FUNCTION}|\text{POSITION}]$ and $H[\text{FUNCTION}|\text{CASE}]$ from the data set by

[4] Since some datings are approximate (e.g. a manuscript can be dated by the first quater of a century), in some cases we had to choose an arbitrary date within the attributed period.

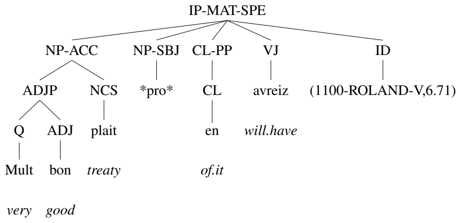
[5] We ran the query twice: on clauses with a finite verb and a subject (whether or not they contained a direct object) and on clauses with a finite verb and a direct object (whether or not they contained a subject). We then merged the two sets of coding strings where each line ended up corresponding to a subject or a direct object token.

[6] In our sample there were no cases of discontinuous subject constituents headed by a noun whereby one part of the constituent would precede the verb and the other one follow, thus creating ambiguity for determining the precedence relation. Thanks to an anonymous reviewer for bringing up this potentially problematic issue.



DATE	*1100*
FUNCTION	*sbj*
POSITION	*pre* (NP-SBJ precedes VJ)
CASE	*nom* (ending *s* in singular)

Figure 1: Coding for subject "The king Marsile holds it"



DATE	*1100*
FUNCTION	*obj*
POSITION	*pre* (NP-ACC precedes VJ)
CASE	*acc* (no nominative ending)

Figure 2: Coding for object "You will have a very good treaty out of this."

maximum likelihood estimation. Note that this partition has been defined with the goal of avoiding data sparsity issues and ensuring that the actual counts in the data set are sufficiently high.[7]

2.4 Results

Entropy measures for the 1st declension are illustrated in Fig. 3, where high entropy corresponds to "weak" and low entropy to "strong" signals. For instance, high conditional entropy of FUNCTION given POSITION means that the probability for an argument in the preverbal position of being a subject was similar to that of being an object, while low entropy indicates a substantial difference. Overall, we can see that the entropy of FUNCTION given POSITION goes down, whereas the entropy of FUNCTION given CASE goes up.

A note is in order concerning an apparent zig-zag of the case signal measure, which, as it were, descends at the XIII c. and then goes back up at the XIV c. Upon closer examination, it turns out that the higher (compared to the following period) entropy in the XII c. is due to the lexical properties of one text, namely, *Li Quatre Livre des Reis*. Here among accusative marked arguments there are 496 objects and 398 subjects. However, among the latter, there are 215 tokens of the name *David*. In the corpus this name appears in the nominative form, *Davids*, only

[7] In other words, we do not face the same kind of estimation problems that are reported for instance by [2]. We also illustrate this in the next few sections by reporting error bars, on the plots, computed by statistical bootstrapping.

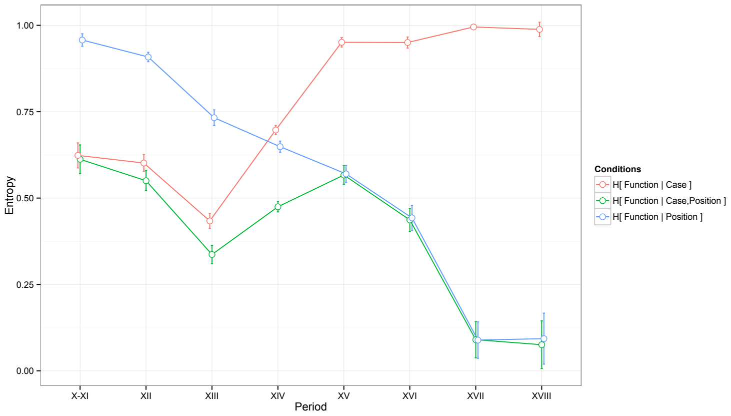


Figure 3: Morphological case and position as signals of grammatical function

twice, in the chronicles of Jean Froissart dated from approximately 1370. Given that proper nouns may have different morphological behaviour then common noun, we also did entropy estimations on the set of common noun only, Fig. 4.

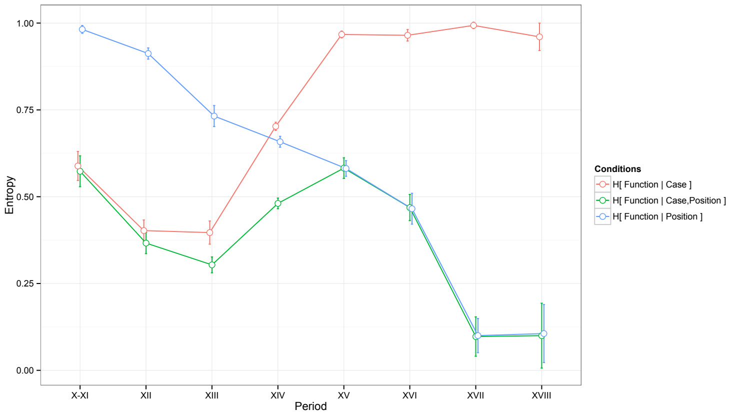


Figure 4: Morphological case as a signal of grammatical function for common nouns

Coming back to the general picture, if we assume that the purported "trade-off" in signal strength is immediate, we then expect that when the position-signal was still very weak (entropy around 0.9), the case-signal should have been strong in order to efficiently mark subject/object distinction. Instead, what we find is a weak case signal in the earliest periods of MF (entropy around 0.6). That is, it appears that if the weakening of case signal was indeed the trigger of the word order changes, the effect was not immediate. That there can be a temporal lag between morphological changes and their possible syntactic consequences has been suggested in the studies of the relation between the impoverishment of verbal inflection and the disappearance of verbal movement in Germanic languages (e.g. [25]).

Another question, however, is why it should be the position and not a set of new morphological markers which replaces the lost case, and, specifically, why subjects occupy preverbal and not postverbal position. Below we investigate the hypothesis that nominal subjects became strictly preverbal by analogy with pronominal subjects, whose rate soared in MF, ([4], [18]), following the syncretisation of verbal subject agreement.

3 Verbal inflection and loss of pro-drop

Part of the morphological impoverishment of MF was the spread of the subject agreement ending *e* from the 3rd to the 1st person singular in verb forms of the traditional 1st conjugation class (with *-er* infinitives) in indicative and subjunctive moods of the present tense ([11, 200,207]) (*aim* '(I) love' becomes *aime*, as in *il aime* 'he loves'), as well as the spread fro the ending *s* from the 2nd to the 1st person singular in verb forms of the traditional 2nd conjugation class (with *-ir*, *-oir* and *re* infinitives). A non-syncretised paradigm identifies the subject's person right at the position of V, which is impossible with an ambiguous *e*, given the possibility of pro-drop and a flexible word order. However, if the pronominal subject is always overt (i.e. there is no pro-drop), identification of the subject's person is more efficient: pronouns in MF are most often preverbal and unambiguous as to their grammatical role.

The disappearance of pro-drop in MF has been linked to the impoverishment of the verbal inflection ([21], [13]), but there has been no quantificational studies of the data bearing on the possible connection. We examine the two phenomena, again, in terms of entropy measures. In order to estimate the efficiency of verbal inflection for identification of subject's person we define a binary variable PERSON with sample space $\{1\text{st}, 2\text{nd}, 3\text{rd}\}$ and estimate its entropy given endings *e* and *s*. Most likely syncretisation extended beyond these endings in oral language affecting all final stops and fricatives and making all endings phonologically indistinguishable except for 1st and 2nd person plural. However, due to the unavailability of oral data, we have to approximate this process by focusing on the fate of *e* and *s*, which can be quantified.

3.1 Data extraction

We extracted all clauses with 1st conjugation verb forms ending in *e* or with 2nd conjugation verb forms ending in *s* and with an overt nominal or pronominal subject (total of 3,202). This allowed us to estimate how good the two endings were to predict subject's person. Below we explicate the coding procedure. The variables we coded for are as follows:

1. DATE is extracted as in section 2.3.
2. CONJUGATION of the verbal form:[8]
   - *first* if the form belonged to the first conjugation.
   - *first* if the form belonged to the second conjugation.
3. ENDING of the verbal form. The codes were assigned corresponding to endings of verbal forms, such as:[9]
   - The code *e* was assigned if the verbal form ended in *e*, *ë*, é or è.
   - The code *s* was assigned if the verbal form ended in *s*, *z* or *x*.[10]
4. PERSON of the subject: *first*, *second*, or *third*.[11]

We estimated the conditional entropy $H[\text{PERSON}|\text{ENDING}]$ from the data set by maximum likelihood estimation. In order to track the evolution of pro-drop, we estimated the entropy $H[\text{SUBJECT}]$ of the variable SUBJECT, which coded all clauses with a finite verb and either a null or a pronominal subject for the presence/absence of an overt pronominal subject (*yes*, *no*).[12]

3.2 Results

The results are illustrated in Fig. 5. Entropy of subject's person given ending predictably increases and eventually goes up to 1, meaning that *e* and *s* progressively become indiscriminate with respect to the person of the subject, reaching maximum ambiguity by the end of the MF period.

At the same time, entropy of SUBJECT goes down, that is, the probability of having an overt pronominal subject becomes progressively greater than not having one. We also see that entropy of subject's person given ending had already been well above 0 when entropy of *Subject* was still 1, meaning, in our model, that syncretisation precedes the decline of pro-drop, which corroborates (but does not prove, of course) the hypothesis that the former triggered the latter.

[8] Similarly to our treatment of nominal declensions, we extracted verbal forms from the clauses with an overt nominal or pronominal subject and listed them separately according to their conjugation type.

[9] This is not an exhaustive list of endings we used in our coding, but in this paper we are interested only in *e* and *s*.

[10] Our query made sure to avoid confusion between other endings with final *s*, *z*, x (such as 2nd and 1st person plural endings *ez*, *ons* etc.) and the relevant endings.

[11] We extracted all pronominal forms from the corpus and classified them by person.

[12] We excluded from our counts coordination structures with subject ellipsis, since this phenomenon persists in Modern French as well and is therefore irrelevant for the question of the evolution of overt pronominal subjects.

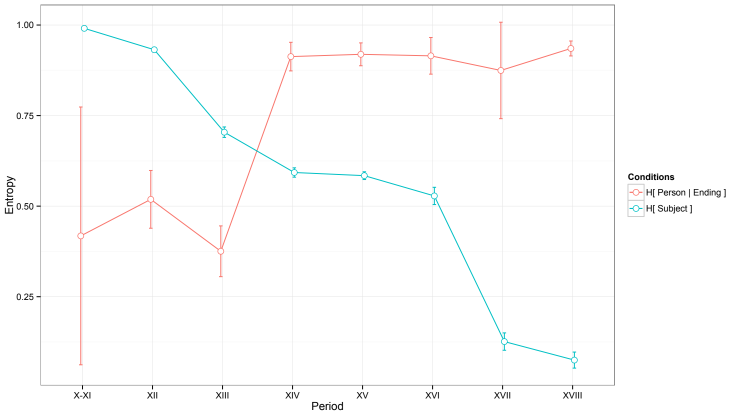


Figure 5: Pro-drop and verbal endings *s* and *e* as signals of subject's person

4 Pronominal and nominal subjects: analogy

We now evaluate the hypothesis that the massive appearance of overt pronominal subjects, almost always preverbal, triggered an analogical change in the syntax of nominal subjects which progressively became preverbal. First, we need to establish the fact that the growing rate of utterances with overt subjects is due to the emergence of overt pronominal subjects, whereas the rate of nominal subjects was declining. To that end, we coded the corpus for the following variables.[13]

1. DATE is extracted as in section 2.3.

2. PRONOUN received value *yes* if an utterance contained an overt pronominal subject and *no* otherwise.

3. NOUN received value *yes* if an utterance contained a non-pronominal subject and *no* otherwise.

4. POSITION of the subject with respect to the finite verb (*pre* vs. *post*).

Fig. 6 shows that the probability of having a non-pronominal subject was slowly going down from 25% to 0.05%, whereas the probability of an overt pronominal subject raised from 41% to 92%.

Pronominal subjects in MF are overwhelmingly preverbal. For instance, in X–XI cc. there was about 56% of preverbal nominal subjects (465 out of 832) whereas among pronominal subject the rate was 68% (1039 out of 1534). On the hypothesis about an analogical change in the syntax of nominal subjects, we compare the profile of the emergence of overt pronominal subjects and the fixation

[13] We excluded relatives clauses, imperatives, and wh-questions because of their idiosyncratic subject syntax, as well as coordination structures with subject ellipsis.

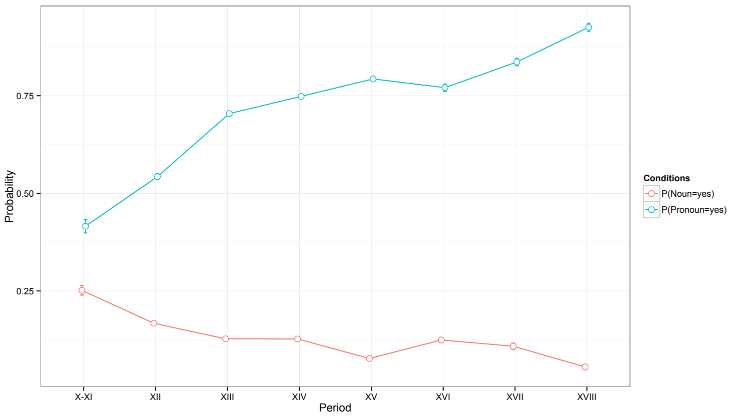


Figure 6: Pronominal and non-pronominal subjects

of nominal subjects in the preverbal position. Fig. 7 shows the probability of an overt pronominal subject calculated on the sample of clauses with a pronominal subject or without an overt subject, P(SUBJECT = *yes*), and the probability of *non-pronominal subjects* being preverbal, P(POSITION = *pre*, NOUN = *yes*). The two measures are significantly correlated (Pearson's r = 0.82, p = 0.01).

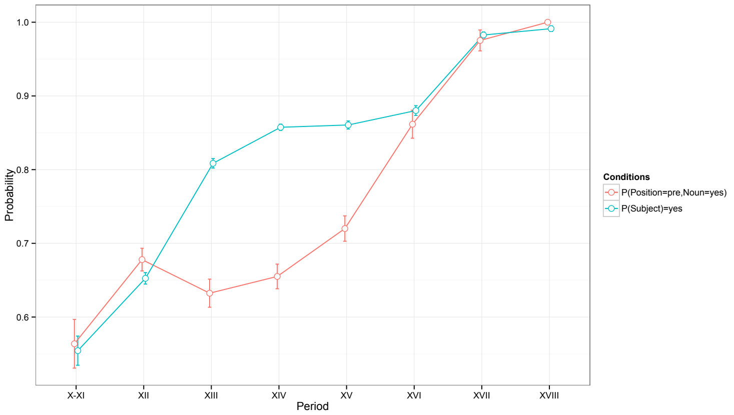


Figure 7: Preverbal non-pronominal subject and overt pronominal subject

5 Conclusions

In this paper we showed, first, that the loss of case and the fixation of argument positions, if taken as signals of grammatical functions, are in a tradeoff relation, assuming that a tradeoff does not have to be immediate. That is, the position signal is still very weak at the time when case signal is already imperfect. It must be noted, however, that due to the lack of data prior to X c., we cannot estimate whether morphological case was ever a perfect function signal (i.e. completely unambiguous). Second, our results suggest that a similar tradeoff relation was holding between the degree of unambiguity of verbal endings and the rate of expression of pronominal subjects. As a side note, one cannot help noticing the striking similarity between the temporal profiles of the two morphological phenomena, case and endings signals, which we will have to leave to future research. Third, we found a strong correlation between the replacement of pro-drop by overt pronominal subjects and the migration of non-pronominal subjects to the preverbal position. A correlation does not of course entail causality, but the results suggest that the two were related in a highly non-accidental manner. Parsed treebanks made it possible for us to develop with Information Theoretic expressions for morphological and syntactic phenomena thereby making them comparable on the diachronic plane, which is a novel contribution.

References

[1] Lucien Foulet. *Petite syntaxe de l'ancien français*. Champion, troisième édition revue. Réédition 1982, Paris, 1928.

[2] Richard Futrell, Kyle Mahowald, and Edward Gibson. Quantifying word order freedom in dependency corpora. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 91–100, 2015.

[3] Aafke Hulk and Ans van Kemenade. Verb second, pro-drop, functional projections and language change. In *Clause structure and language change*, pages 227–256. Oxford University Press, 1995.

[4] Georg A. Kaiser. Losing the null subject. A contrastive study of (Brazilian) Portuguese and (Medieval) French. In *Proceedings of the Workshop "Null-subjects, expletives, and locatives in Romance"*, pages 131–156, 2009.

[5] Edward Keenan. Toward a universal definition of 'subject'. In C Li, editor, *Subject and Topic*. Academic Press, New York, 1976.

[6] Paul Kiparsky. The rise of positional licensing. In Ans van Kemenade and Nigel Vincent, editors, *Parameters of morphosyntactic change*, pages 460–494. Citeseer, 1997.

[7] Anthony Kroch and Beatrice Santorini. On the word order of Early Old French, 2014. SinFonIJA 7, Graz, Austria, September 2014.

[8] Marie Labelle. Clausal architecture in Early Old French. *Lingua*, 117(1):289–316, 2007.

[9] Marie Labelle and Paul Hirschbühler. Changes in clausal organization and the position of clitics in Old French. In M. Batllori, M.L. Hernanz, C. Picallo, and F. Roca, editors, *Grammaticalization and Parametric Variation*, pages 60–71. Oxford University Press, Oxford, 2005.

[10] Adam Ledgeway. *From Latin to Romance: Morphosyntactic typology and change*. Oxford University Press, 2012.

[11] Christiane Marchello-Nizia. *Histoire de la langue française aux XIVe et XVe siècles*. Dunod, Paris, 1992.

[12] Christiane Marchello-Nizia and Magali Rouquier. De (S)OV à SVO en français: où et quand? L'ordre des constituants propositionnels dans la Passion de Clermont et la Vie de saint Alexis. In Monique Dufresne, editor, *Constructions en changement. Hommage à Paul Hirschbüler*, pages 111–155. Presses de l'Université de Laval, 2012.

[13] Eric Mathieu. Stylistic fronting in old french. *Probus*, 18(2):219–266, 2006.

[14] Thomas McFadden. *The positon of morphological case in the derivation: A study on the syntax-morphology interface*. PhD thesis, University of Pennsylvania, 2004.

[15] Corpus MCVF annoté syntaxiquement, sous la direction de France Martineau, avec Paul Hirschbühler, Anthony Kroch et Yves Charles Morin, 2010.

[16] Penn Supplement to the MCVF Corpus by Anthony Kroch and Beatrice Santorini, 2010.

[17] Harm Pinkster. *Latin Syntax and Semantics*. Routledge, New York, 1990.

[18] Sophie Prévost. Emergence and development of personal pronoun subjects: study of a bilingual Latin-old French corpus, To appear.

[19] Ian Roberts. *Verbs and Diachronic Syntax: A Comparative History of English and French*. Kluwer, Dordrecht, 1993.

[20] Ian Roberts. Directionality and word order change in the history of English. In Ans Van Kemenade and Nigel Vincent, editors, *Parameters of morphosyntactic change*, pages 397–426. Cambridge University Press, 1997.

[21] Bernhard Wolfgang Rohrbacher. *The Germanic VO languages and the full paradigm: a theory of V to I raising*. PhD thesis, University of Massachusetts, Amherst, 1994.

[22] Edward Sapir. *Language, an introduction to the study of speech.* Harcourt, Brace and Co., New York, 1921.

[23] Lene Schøsler. La déclinaison bicasuelle de l'ancien français: son rôle dans la syntaxe de la phrase, les causes de sa disparition. In *Études romanes de l'Université d'Odense*, volume 19. Odense University Press, 1984.

[24] Achim Stein and Sophie Prévost. Syntactic annotation of medieval texts: the Syntactic Reference Corpus of Medieval French (SRCMF). In P. Bennett, M. Durrell, S. Scheible, and R. Whitt, editors, *New Methods in Historical Corpus Linguistics. Corpus Linguistics and International Perspectives on Language*, volume 3, pages 75–82. CLIP, Tübingen, 2013.

[25] John D. Sundquist. *Morphosyntactic change in the history of the Mainland Scandinavian languages.* PhD thesis, Indiana University, 2002.

[26] Barbara Vance. *Syntactic change in medieval French.* Studies in Natural Language and Linguistic Theory. Kluwer, Dordrecht/Boston/London, 1997.

[27] Theo Vennemann. An Explanation of Drift. In Charles Li, editor, *Word Order and Word Order Change*, pages 269–305. University of Texas Press, Austin, 1975.

[28] Laurie Zaring. On the nature of OV and VO order in early Old French. *Lingua*, 121:1831–1852, 2011.

Parsing Universal Dependency Treebanks Using Neural Networks and Search-Based Oracle

Milan Straka, Jan Hajič, Jana Straková and Jan Hajič jr.

Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University in Prague
E-mail: `{straka|hajic|strakova|hajicj}@ufal.mff.cuni.cz`

Abstract

We describe a transition-based, non-projective dependency parser which uses a neural network classifier for prediction and requires no feature engineering. We propose a new, search-based oracle, which improves parsing accuracy similarly to a dynamic oracle, but is applicable to any transition system, such as the fully non-projective `swap` system, contrary to dynamic oracles, which are specific for each transition system and usually quite complex. The parser has excellent parsing speed, compact models, and achieves high accuracy without requiring any additional resources such as raw corpora. We tested it on all 37 treebanks of the Universal Dependencies project. The C++ implementation of the parser is being released as an open-source tool.

1 Introduction

Transition-based systems were proposed by Yamada and Matsumoto [28] and Nivre [16]. Greedy transition-based parsers are very efficient while achieving reasonably high accuracy, allowing to parse large volumes of data.[1]

An *oracle* is used at training time to map parser configurations to optimal transitions given a gold tree. A classifier is then trained to emulate the oracle predictions.

Initially, transition-based parsers used *static oracles*, which are defined only for configurations from which the complete gold tree can be reached. Recently, Goldberg and Nivre [9, 10], Goldberg et al. [11], Gómez-Rodríguez et al. [13] and Gómez-Rodríguez and Fernández-González [12] improved accuracy of transition-based parsers by utilizing a *dynamic oracle*, which is defined for any parser configuration and predicts transitions leading to a tree most similar to the gold one. Such a dynamic oracle affects only the training speed, not parsing speed. However, a

[1] Beam search can improve parsing accuracy but at a substantially lower speed, cf. e.g. Zhang and Nivre [31].

dynamic oracle is usually more complicated than a static one; for example, the dynamic oracle of Gómez-Rodríguez et al. [13] for a restricted non-projective system has $O(n^8)$ complexity.

In this paper we consider a new *search-based oracle*, which resembles the dynamic oracle in terms of predicting transitions from any parser configuration. However, a search-based oracle utilizes only the classifier being trained, which makes it applicable to any transition system with a static oracle only. Still, parsing accuracy of a search-based oracle is comparable to the dynamic oracle.

Inspired by recent success of distributed word representations in NLP, e.g. in POS tagging (Collobert et al. [3]), machine translation (Devlin et al. [6]), constituency parsing Socher et al. [25] and projective dependency parsing (Chen and Manning [2]), we train a neural network (NN) classifier predicting transitions in a transition-based parser. We utilize the search-based oracle allowing the `swap` operation and thus more accurate fully non-projective parsing. We train our parser on all 37 Universal Dependencies (UD) treebanks version 1.2, showing that high accuracy can be achieved by the new search-based oracle and using a neural network classifier even without additional raw corpora.

The main contributions of this work are:

- a novel search-based oracle which can be used with any transition system, improving the parsing results considerably, comparably to using a dynamic oracle (Sect. 4):
    - notably, the search-based oracle can be applied to the non-projective transition system with the `swap` operation, which enables fully non-projective parsing;
    - the search-based oracle can be used even on top of a dynamic oracle, further improving accuracy;
- a NN-based parser with better accuracy for most of the UD treebanks and substantially improved speed for all of them, while keeping models compact (Sect. 3);
- an open-source C++ parser implementation[2] and parsing models for all 37 treebanks of Univeral Dependencies Treebanks version 1.2 [27].

2 Transition-Based Dependency Parsing

Transition-based dependency parsing computes the dependency tree for a sentence by starting in an initial configuration and performing a sequence of transitions reaching some terminal configuration.

One of the most popular transition systems is the projective stack-based arc-standard system by Nivre [17], which we denote as `stack`. This system employs three types of transitions: *left_arc*$_l$ and *right_arc*$_l$, which add a dependency arc with label l, and *shift*, which adds the next input word.

There are also several transition systems that allow parsing of non-projective trees. Attardi [1] introduced transitions to the `stack` system adding dependency

[2] `http://hdl.handle.net/11234/1-1573`

arcs between non-adjacent subtrees. Here we consider a restriction of the original Attardi parser described for example in Gómez-Rodríguez et al. [13], which we denote as `arc2`. The `arc2` system extends the `stack` system by adding transitions *left_arc_2$_l$* and *right_arc_2$_l$* which add dependency arcs between non-adjacent nodes. Although only some non-projective trees can be obtained by such transitions, Attardi in [1] notes that the `arc2` system is sufficient to handle almost all cases of non-projectivity in the training data.

The truly non-projective transition system which we call `swap` was proposed by Nivre [18]. It extends the `stack` system by adding the *swap* transition for reordering two nodes. Nivre et al. [20] show that any non-projective tree can be reached while keeping the expected time linear.

3 Neural Network Classifier

The architecture of the neural network classifier is similar to that described in Chen and Manning [2].

The input to the network consists of several nodes representing words in the tree being built. Following Zhang and Nivre [31] and Chen and Manning [2], we use a rich set of up to 18 nodes as input: top 3 nodes on the stack, top 3 nodes on the buffer, the first and second leftmost/rightmost children of the top 2 nodes on the stack, and leftmost of leftmost and rightmost of rightmost children of the top 2 nodes on the stack.

Each node is represented using distributed representations of its form, its POS tag and its arc label; the latter only if it has already been assigned.

In the Universal Dependency treebanks, there are three token fields connected to part-of-speech: UPOSTAG (universal part-of-speech tag), XPOSTAG (language-specific part-of-speech tag, which is not present in many treebanks) and FEATS (list of morphological features further refining the universal part-of-speech tag). We use both UPOSTAG and FEATS fields, which improves results considerably, compared to using only UPOSTAG.

The input layer is connected to a hidden layer with *tanh* activation. The output layer has a node for every transition and uses *softmax* activation.

3.1 Distributed Word Representations

POS-tag, FEATS and arc-label embeddings are initialized randomly and trained together with the network. Form embeddings are pre-trained using `word2vec` (Mikolov et al. [14]), employing the Skip-gram model with negative sampling.[3] We pre-train the embeddings only on the treebank data, to show that the resulting parser works with high accuracy without additional resources, which might be hard to obtain for some languages. Because all form embeddings are currently in

[3] The exact options for `word2vec` were the following: `-cbow 0 -size 50 -window 10 -negative 5 -hs 0 -sample 1e-1 -iter 15 -min-count 2`

the training data, we train them further together with the network, yielding a small accuracy improvement.

All forms appearing only once in the training data are replaced by a unique unknown-word token. Its embedding is then used for OOVs during parsing.

3.2 Training the Classifier

We train the neural network by stochastic gradient descent (Robbins and Monro [22]) with mini-batches of size 10, minimizing cross-entropy loss with L_2-regularization. We employ exponential learning-rate decay. For all treebanks, we use form embeddings of dimension 50, POS tag, FEATS and arc label embeddings of dimension 20, and a 200-node hidden layer. Other hyperparameters[4] are determined based on the development portion of the treebanks and the best combination is used.

We would like to note that although we tried several advanced neural network training techniques, notably AdaGrad (Duchi et al. [7]), dropout (Srivastava et al. [26]), cube activation function (reported to improve performance by Chen and Manning [2]), or AdaDelta (Zeiler [29]), none helped and the best accuracy was obtained by the basic mini-batched SGD.

3.3 Improving Classification Speed

We have used several techniques to improve the transition classification speed, which in turn directly determines parsing speed. Similar to Devlin et al. [6], we pre-computed the hidden layer increments for all embeddings and all input layer positions. We also compute the *tanh* using table lookup (except during training in order to obtain accurate gradients) and we do not normalize the output layer during parsing.

4 Search-Based Oracle

When training the classifier using a static oracle, the same sequence of transitions is always used for every sentence. In other words, the classifier is trained only on transition sequences which do not contain any incorrect transitions. If the classifier is then used to parse a sentence and makes an error, it is difficult for it to recover from this error, because the classifier never encountered such situation in training data.

The dynamic oracle (Goldberg and Nivre [9]) improves the situation by being able to provide the best transition from an arbitrary configuration, even if some incorrect transitions have already been performed. When training with a dynamic oracle, usually an exploration policy parametrized by k and p is used to determine

[4] To be specific, the hyperparameters are: number of training iterations (between 5 and 10), initial learning rate (between 0.01 and 0.02), final learning rate (between 0.001 and 0.005) and L_2-regularization (between 0.1 and 0.5).

which transition to follow: during the first k iterations the oracle transition is always chosen (as with the static oracle), but later the oracle transition is chosen only with probability $1-p$, using the (possibly incorrect) classifier prediction otherwise. Consequently, the classifier is being trained on sequences of transitions which it predicts itself.

The main idea behind our search-based oracle is to approximate the dynamic oracle by the current state of the classifier being trained. This approach is inspired by the Searn algorithm of Daumé III et al. [4], a method for reducing error propagation during structured prediction.

Specifically, when determining the transition to follow for a given parser configuration with the search-based oracle, we perform every applicable transition in sequence and for such transition we use the classifier being trained to parse the rest of the tree (by following the predicted transition in every step). We then choose such transition from the original configuration which results in a dependency tree with the highest attachment score.

As many transitions differ only in the label of the arc being added, to improve oracle speed, we employ the following heuristic: when choosing a transition to follow, we consider only those arc-adding transitions that assign the label appearing in the gold tree. This effectively reduces the number of possible transitions from tens to at most five (e.g., from 96 to 4 transitions in the `swap` system for English).

When training with the search-based oracle, we have to make sure that the original oracle is employed frequently enough, because the original oracle is the only way of utilizing gold data. Therefore, unlike with the dynamic oracle, where the exploration policy alternates between the dynamic oracle prediction and classifier prediction on every transition, we use the following policy: after training on *interval* sentences with the static oracle, we train one sentence with the search-based oracle. The *interval* becomes another hyperparameter of our system tuned on the development part of the treebank (we consider *interval* between 8 and 10).

The training time of a search-based oracle is naturally higher than the training time of a static oracle, because one prediction of a search-based oracle takes time linear in the size of the sentence being parsed. For the values of *interval* used, the training time of a search-based oracle is 2-3 times worse than training time of a static oracle alone. This is comparable to a dynamic oracle for the `stack` system, which is reported to have training time slower by a factor of 2.3 when using a dynamic oracle instead of a static one. Also note that this slowdown applies only to training, parsing speed of the trained classifier is exactly the same for static, search-based and dynamic oracles.

Interestingly, our search-based oracle can be combined not only with a static oracle, but also with a dynamic oracle, yielding accuracy improvements for the dynamic oracle, too.

5 Experiments

We evaluate parser accuracy on treebanks from the Universal Dependencies project, which seeks to develop cross-linguistically consistent treebank annotation for many languages. The annotation scheme is based on the universal Stanford dependencies (de Marneffe et al. [5]), the Google universal part-of-speech tags (Petrov et al. [21]), and the Interset interlingua for morphosyntactic features (Zeman [30]).

Namely, we use the 37 dependency treebanks of Univeral Dependencies Treebanks version 1.2 [27]. Four basic statistics of each treebank are presented in columns 2 and 3 of Table 1.

The results of our parser with the `stack`, `swap` and `arc2` systems are presented in the rest of Table 1. We report unlabeled attachment scores (UAS) and labeled attachment scores (LAS), excluding punctuation, computed using MaltEval (Nilsson and Nivre [15]). We show the results with a static oracle only and using our search-based oracle. For comparison, we also present results of a dynamic oracle (we implemented the dynamic oracle for the `stack` system from Goldberg et al. [11] and used it with the same classifier as the search-based oracle) and results of a search-based oracle used on top of a dynamic one.[5]

We also report results of MaltParser (Nivre et al. [19]), a greedy transition-based parser using liblinear (Fan et al. [8]) for optimization. We used MaltParser version 1.8.1. with default options and feature templates, changing the transition system (using `stackproj` and `stacklazy` as `stack` and `swap`, respectively), number of iterations (computed using treebank size), and passing a concatenation of UPOSTAG and FEATS fields as POS tags to use. We used MaltParser because it is a transition-based parser that implements many transition systems (including non-projective) which we wanted to compare with, and is very fast. It is therefore similar to our parser, in contrast to a slow parser achieving higher accuracy.

We also report parsing speed and model size of the `swap` parser. Parsing speed was measured on an Intel Pentium G850 2.9GHz CPU with 4GB RAM and it does not include model loading time.

5.1 Results

Comparing our static-oracle-only parser to MaltParser, our parser has better accuracy, achieving on average 6.2% relative error reduction in UAS and 6.7% in LAS. Our parser produces models on average half the size of MaltParser's (with models 4-5 times smaller for Czech, Ancient Greek, and Latin), and it is faster (20-30k words/s, on average 3.6 times faster than MaltParser).

The search-based oracle parser is clearly superior to the static oracle parser, achieving additional 4.3% relative error reduction in UAS and 3.6% relative error

[5] We did not implement any other dynamic oracle, because the dynamic oracle for the `arc2` system is very complicated with its $O(n^8)$ complexity, no dynamic oracle for the `swap` system is known to the best of our knowledge, and the recent dynamic oracle for the fully non-projective Covington parser of Gómez-Rodríguez and Fernández-González [12] uses a quite different transition system.

Language	Size	Non-proj.	Static oracle			Search-based oracle			DynO	SB+DO	MaltParser	
	Words	Non-proj. edges	Stack UAS	Swap UAS	Arc2 UAS	Stack UAS	Swap UAS	Arc2 UAS	Stack UAS	Stack UAS	Stack UAS	Swap UAS
	Sentences	Non-proj. sentences	Stack LAS	Swap LAS	Arc2 LAS	Stack LAS	Swap LAS	Arc2 LAS	Stack LAS	Stack LAS	Stack LAS	Swap LAS
Ancient	244 993	9.78%	58.6	66.2	66.5	64.2	**69.3**	68.5	66.4	67.7	*55.1*	*65.3*
Greek	16 221	63.22%	53.0	60.6	60.9	58.5	**63.9**	62.8	60.5	62.0	*49.4*	*59.4*
Ancient	206 966	5.95%	72.3	75.7	74.8	74.4	**76.1**	75.5	75.8	75.9	*69.7*	*73.4*
Greek–PROIEL	16 633	39.48%	67.0	70.6	69.6	69.2	**71.3**	70.5	70.7	71.0	*64.5*	*68.7*
Arabic	282 384	0.33%	79.9	79.8	80.2	80.4	80.6	**80.7**	78.2	79.4	*80.1*	*79.7*
	7 664	8.19%	74.6	74.7	75.3	75.5	**75.8**	75.7	73.4	74.7	*74.6*	*74.3*
Basque	121 443	4.95%	77.0	78.3	78.4	78.2	79.2	79.6	79.9	**80.6**	*74.7*	*77.3*
	8 993	33.74%	71.9	73.1	73.2	73.5	74.3	74.5	75.2	**76.0**	*68.9*	*71.5*
Bulgarian	156 319	0.21%	90.2	90.7	90.9	91.1	91.2	**91.5**	90.5	91.2	*89.2*	*89.5*
	11 138	2.83%	84.8	85.5	85.7	86.0	86.1	**86.2**	85.3	86.0	*83.2*	*83.6*
Croatian	87 765	0.46%	81.1	80.8	80.2	82.1	82.4	81.3	**82.7**	82.0	*77.4*	*78.5*
	3 957	7.48%	73.9	73.6	72.5	75.2	**75.3**	74.4	74.8	74.7	*69.7*	*70.9*
Czech	1 506 490	0.93%	86.7	87.9	87.8	87.7	88.0	**88.2**	87.2	87.5	*85.2*	*86.3*
	87 913	12.58%	83.2	84.3	84.4	84.3	84.7	**84.8**	83.8	84.1	*81.3*	*82.4*
Danish	100 733	1.97%	81.8	82.5	82.9	82.7	82.8	**83.3**	82.6	**83.3**	*80.1*	*81.4*
	5 512	22.84%	78.0	79.1	79.3	79.2	79.2	**80.0**	78.8	79.6	*75.5*	*76.8*
Dutch	200 654	4.10%	74.6	75.8	76.2	76.0	**77.5**	77.1	76.0	75.7	*71.9*	*75.8*
	13 735	30.87%	70.8	72.0	71.8	72.0	**73.8**	73.1	72.1	72.3	*67.9*	*71.2*
English	254 830	0.48%	86.7	86.5	86.9	87.4	87.2	87.3	87.3	**87.7**	*86.3*	*86.5*
	16 622	4.96%	84.2	83.8	84.2	**84.7**	84.5	84.5	84.5	**84.7**	*82.9*	*83.2*
Estonian	9 491	0.08%	85.0	85.3	86.0	87.4	86.5	86.3	86.4	86.2	*86.4*	***88.1***
	1 315	0.61%	81.7	81.9	83.0	83.2	82.8	83.1	83.1	83.0	*83.8*	***85.7***
Finnish	181 022	0.74%	80.4	81.2	81.1	81.5	81.7	81.6	82.7	**83.5**	*81.0*	*80.8*
	13 581	7.68%	77.0	77.9	77.6	78.2	78.3	78.6	79.2	**80.2**	*76.9*	*77.0*
Finnish–FTB	159 829	1.09%	80.3	80.1	80.0	81.3	81.0	80.4	81.6	**82.3**	*79.6*	*80.1*
	18 792	6.78%	77.2	76.9	76.6	78.1	78.0	77.3	78.0	**79.1**	*75.8*	*76.3*
French	401 491	0.83%	84.2	85.0	84.7	85.2	**85.5**	85.2	84.5	85.0	*83.3*	*83.4*
	16 446	12.45%	80.4	81.2	81.1	81.5	**81.7**	81.4	80.6	81.2	*78.8*	*78.8*
German	298 242	0.90%	82.3	82.6	83.0	83.3	83.3	83.1	83.2	**84.4**	*81.3*	*82.2*
	15 894	12.08%	76.9	77.1	77.6	78.0	78.0	77.6	77.6	**78.8**	*75.2*	*75.8*
Gothic	56 128	3.86%	76.2	76.1	76.2	78.3	77.4	77.9	78.0	**78.5**	*75.2*	*76.2*
	5 450	23.85%	70.5	70.4	70.7	72.2	71.4	72.4	72.1	**73.0**	*69.1*	*70.5*
Greek	59 156	1.95%	81.3	81.7	82.5	**82.9**	82.5	**82.9**	82.2	82.8	*79.0*	*80.6*
	2 411	27.87%	78.4	78.4	79.2	79.3	79.1	79.6	79.0	**79.8**	*75.2*	*77.1*
Hebrew	158 855	0.00%	85.1	86.0	85.9	86.0	**86.2**	86.1	85.6	85.8	*83.2*	*83.1*
	6 216	0.00%	80.6	81.1	81.3	81.6	**81.9**	81.4	81.2	81.8	*78.5*	*78.4*
Hindi	351 704	0.76%	92.5	93.3	93.0	93.3	93.7	93.6	93.8	**93.9**	*89.4*	*89.5*
	16 647	13.60%	89.3	90.0	89.7	90.1	90.5	90.3	**90.6**	**90.6**	*84.5*	*84.6*
Hungarian	26 538	2.09%	79.9	80.3	79.0	80.4	80.6	81.2	81.3	**81.9**	*78.2*	*79.1*
	1 299	25.17%	74.2	74.3	72.9	75.1	75.5	75.6	75.8	**77.5**	*72.7*	*74.0*
Indonesian	121 923	0.13%	83.1	83.1	**83.3**	**83.3**	**83.3**	**83.3**	82.1	82.4	*81.7*	*81.8*
	5 593	1.93%	77.8	77.6	78.0	77.9	**78.2**	77.9	76.7	77.0	*75.8*	*75.9*
Irish	23 686	0.81%	74.6	74.2	73.6	75.2	75.2	75.1	74.4	74.6	***75.4***	*73.8*
	1 020	12.84%	67.4	66.8	66.7	68.1	**68.5**	67.5	68.0	67.7	*67.6*	*66.4*
Italian	271 180	0.32%	90.1	90.0	90.3	90.6	90.6	**90.8**	89.8	90.6	*89.0*	*88.8*
	12 677	3.94%	87.7	87.5	87.8	88.0	88.1	**88.4**	87.3	88.2	*86.4*	*86.2*
Japanese–KTC	267 631	0.00%	85.1	85.2	84.9	85.5	**85.7**	**85.7**	85.1	85.3	*84.2*	*84.1*
	9 995	0.00%	75.1	75.0	74.8	**75.5**	75.3	75.3	75.1	75.2	*72.9*	*73.3*
Latin	47 303	7.13%	58.2	57.2	57.9	59.2	59.2	58.3	**61.1**	60.7	*58.1*	*57.2*
	3 269	46.22%	49.8	50.4	50.6	51.7	52.0	51.0	53.6	**53.9**	*50.2*	*50.1*
Latin–ITT	259 684	3.45%	77.2	80.5	79.0	77.8	**80.8**	79.3	79.8	79.5	*72.4*	*76.3*
	15 295	37.20%	73.8	77.5	75.7	74.6	**77.9**	76.2	76.5	76.6	*68.3*	*72.3*
Latin–PROIEL	165 201	5.22%	73.4	74.3	75.2	74.6	75.2	76.1	76.1	**76.6**	*70.0*	*72.5*
	14 982	30.09%	68.3	69.3	70.1	69.5	70.3	71.0	70.8	**71.5**	*64.8*	*67.7*
Norwegian	311 277	0.60%	89.2	89.2	89.7	89.8	90.0	**90.1**	89.7	**90.1**	*88.9*	*88.9*
	20 045	7.70%	86.8	86.8	87.4	87.7	87.7	**87.8**	87.3	**87.8**	*85.8*	*86.0*
Old Church	57 507	3.71%	81.0	82.6	82.2	82.1	**83.3**	83.0	82.6	82.8	*80.1*	*82.0*
Slavonic	6 346	21.57%	75.4	77.8	76.9	77.0	**78.0**	77.9	77.5	77.9	*75.0*	*77.2*
Persian	152 871	0.38%	83.8	83.1	83.5	84.5	84.2	84.6	84.8	**85.0**	*80.8*	*80.8*
	5 997	5.14%	80.2	79.8	80.0	81.1	80.8	81.2	81.3	**81.5**	*77.2*	*77.2*
Polish	83 571	0.04%	88.3	88.7	88.2	89.0	89.0	89.3	**89.8**	89.5	*87.7*	*87.3*
	8 227	0.32%	84.1	84.6	83.8	84.8	84.5	85.2	**85.5**	85.2	*83.1*	*82.8*
Portuguese	212 545	1.27%	85.8	87.6	87.5	87.5	**88.4**	88.1	86.9	87.5	*84.5*	*85.5*
	9 359	18.44%	82.7	84.6	83.9	84.5	**85.4**	85.0	83.8	84.3	*80.5*	*81.5*
Romanian	12 094	0.89%	75.4	74.5	76.3	76.7	76.9	**77.4**	75.5	76.3	*72.8*	*73.1*
	633	11.37%	61.9	60.9	62.1	62.7	**63.2**	**63.2**	62.2	62.2	*59.5*	*59.6*
Slovenian	140 418	1.11%	86.5	87.3	87.5	87.6	**88.9**	88.1	88.2	88.2	*84.3*	*85.7*
	7 996	13.61%	84.5	85.4	85.4	85.8	**87.0**	86.0	86.1	86.4	*81.9*	*83.4*
Spanish	431 587	0.30%	86.8	86.9	87.1	**87.6**	87.2	87.4	85.7	86.4	*85.4*	*85.2*
	16 013	6.05%	83.6	83.7	83.7	**84.4**	84.1	84.0	82.5	83.4	*81.2*	*81.2*
Swedish	96 819	0.19%	85.3	85.7	85.7	85.9	86.1	86.1	**86.2**	**86.2**	*84.7*	*84.7*
	6 026	2.77%	81.4	81.9	82.0	82.3	**82.5**	**82.5**	82.4	82.4	*80.3*	*80.5*
Tamil	9 581	0.29%	75.8	76.3	76.2	76.6	77.1	75.7	**78.4**	78.0	*78.3*	*78.3*
	600	2.17%	67.1	68.5	67.5	67.9	68.7	67.3	69.6	69.5	***69.7***	*69.4*

Table 1: Parsing accuracy on all treebanks of Universal Dependencies version 1.2. *DynO* stands for dynamic oracle, *SB+DO* for search-based and dynamic oracle.

Language	Size		Swap system		MaltParser	
	Words	Sentences	Speed kw/s	Model MB	Speed kw/s	Model MB
Ancient Greek	244 993	16 221	**27.7**	**3.9**	*9.5*	*23.2*
Ancient Greek–PROIEL	206 966	16 633	**25.9**	**3.4**	*8.7*	*21.2*
Arabic	282 384	7 664	**26.4**	**4.3**	*12.0*	*10.4*
Basque	121 443	8 993	**26.9**	**2.6**	*7.7*	*7.7*
Bulgarian	156 319	11 138	**27.5**	**3.2**	*10.6*	*6.8*
Croatian	87 765	3 957	**23.8**	**2.7**	*8.5*	*7.4*
Czech	1 506 490	87 913	**22.9**	**12.1**	*18.2*	*56.8*
Danish	100 733	5 512	**24.3**	**2.5**	*9.1*	*5.6*
Dutch	200 654	13 735	**26.4**	**3.2**	*11.8*	*9.2*
English	254 830	16 622	**21.8**	**3.2**	*12.5*	*6.3*
Estonian	9 491	1 315	**32.7**	1.6	*2.5*	***0.8***
Finnish	181 022	13 581	**22.9**	**4.1**	*9.5*	*14.5*
Finnish–FTB	159 829	18 792	**31.3**	**3.4**	*9.9*	*11.2*
French	401 491	16 446	**25.1**	4.4	*16.8*	***4.1***
German	298 242	15 894	**27.2**	**4.3**	*15.5*	*4.9*
Gothic	56 128	5 450	**27.6**	**2.0**	*6.7*	*6.0*
Greek	59 156	2 411	**28.3**	**2.1**	*6.4*	*4.5*
Hebrew	158 855	6 216	**22.7**	**2.9**	*11.3*	*8.1*
Hindi	351 704	16 647	**27.7**	**3.2**	*12.7*	*9.6*
Hungarian	26 538	1 299	**20.5**	**1.8**	*3.9*	*3.1*
Indonesian	121 923	5 593	**28.3**	**2.7**	*13.0*	*2.8*
Irish	23 686	1 020	**25.7**	**1.7**	*3.2*	*2.6*
Italian	271 180	12 677	**24.1**	**3.7**	*12.9*	*7.8*
Japanese–KTC	267 631	9 995	**29.3**	1.4	*17.7*	***0.4***
Latin	47 303	3 269	**28.5**	**2.1**	*5.7*	*7.3*
Latin–ITT	259 684	15 295	**26.7**	**2.6**	*11.3*	*15.8*
Latin–PROIEL	165 201	14 982	**25.7**	**3.1**	*8.0*	*18.2*
Norwegian	311 277	20 045	**25.9**	**3.6**	*12.9*	*7.6*
Old Church Slavonic	57 507	6 346	**28.1**	**2.1**	*6.6*	*5.6*
Persian	152 871	5 997	**25.2**	**2.7**	*12.2*	*3.9*
Polish	83 571	8 227	**30.2**	**2.5**	*8.2*	*6.3*
Portuguese	212 545	9 359	**27.4**	**3.4**	*12.4*	*8.2*
Romanian	12 094	633	**21.5**	**1.6**	*2.2*	*1.9*
Slovenian	140 418	7 996	**27.0**	**3.1**	*9.4*	*9.7*
Spanish	431 587	16 013	**26.9**	**4.8**	*13.6*	*12.0*
Swedish	96 819	6 026	**24.9**	**2.3**	*8.5*	*4.3*
Tamil	9 581	600	**31.1**	1.6	*2.3*	***0.9***

Table 2: Parsing speed and model size measured on Universal Dependencies 1.2, using the `swap` transition system.

reduction in LAS.

The dynamic oracle for the `stack` system has very similar results to the search-based oracle for the `stack` system (relative error reduction compared to static oracle is slightly higher for a search-based oracle than for a dynamic oracle), with the search-based oracle being simpler and applicable for any transition-based system. Additionally, the search-based oracle can be used together with the dynamic oracle, yielding further improvement of 2.2% relative error reduction in UAS and 2.3% relative error reduction in LAS on average over the UD 1.2 dataset.

6 Related Work

A neural network based dependency parser was proposed by Chen and Manning [2]. The architecture of our parser is quite similar. However, our parser implements two non-projective transition systems, it utilizes the search-based oracle, and we evaluate performance on 37 treebanks and without form embeddings computed on a large raw corpus.

Since parsing is a structured prediction problem, methods developed to handle error propagation during structured prediction like Searn (Daumé III et al. [4]), SMILe (Ross and Bagnell [23]) or DAgger (Ross et al. [24]) might improve parsing accuracy. The search-based oracle resembles Searn to some extent, as Searn computes the regret of an action by executing the current policy to gain a full sequence of predictions and computing its loss, which is similar to how optimal transitions in the search-based oracle are obtained. On the other hand, the rest of the training with the search-based oracle can be viewed as an approximation of the DAgger algorithm, similarly to the dynamic oracle (Goldberg and Nivre [9]).

Search-based oracle used with the `swap` transition system enables fully non-projective transition based parsing, for which no dynamic oracle existed for a long time. Recently, a dynamic oracle with $O(n)$ complexity for fully non-projective Covington parser was devised by Gómez-Rodríguez and Fernández-González [12]. The Covington parser can be implemented under the transition-based parsing framework (Nivre [17]), but it uses multiple lists of partially processed words and has quadratic worst-case complexity.

7 Conclusions

We have described a non-projective, neural-network based dependency parser Parsito[6] employing a novel, efficient search-based oracle. It has been evaluated on all 37 Universal Dependency treebanks, showing improvements in accuracy and especially in speed. We are releasing the parser and the models as open-source.[7]

The new search-based oracle improves parsing accuracy similarly to a dynamic one (over a static oracle), but it can work with the `swap` system for non-projective parsing (or any other transition system). Even when a polynomial-time dynamic oracle is known, the search-based oracle requires much less effort to implement, and there is still room for improvement (e.g., in the frequency of its use during training). Alternatively, the search-based oracle can be used together with the dynamic oracle to improve parsing accuracy even further.

Our future work includes utilizing character-level embeddings and/or computing word embeddings using large additional corpora. Furthermore, we will experiment with beam search for decoding as an option to improve parsing accuracy at the expense of parsing speed.

[6] Project homepage: `http://ufal.mff.cuni.cz/parsito`

[7] `http://hdl.handle.net/11234/1-1573`

Acknowledgments

This work has been partially supported by and has been using language resources developed, stored and distributed by the LINDAT/CLARIN project of the Ministry of Education of the Czech Republic (project LM2010013). This research was also partially supported by the SVV project number 260 224. We are grateful to the reviewers for comments which helped us to improve the paper.

References

[1] Giuseppe Attardi. Experiments with a multilanguage non-projective dependency parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 166–170, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[2] Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October 2014. Association for Computational Linguistics.

[3] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[4] Hal Daumé III, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine learning*, 75(3):297–325, 2009.

[5] Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, 2014.

[6] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard M. Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1370–1380, 2014.

[7] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12: 2121–2159, July 2011.

[8] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008.

[9] Yoav Goldberg and Joakim Nivre. A dynamic oracle for arc-eager dependency parsing. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 959–976, 2012.

[10] Yoav Goldberg and Joakim Nivre. Training deterministic parsers with non-deterministic oracles. *TACL*, 1:403–414, 2013.

[11] Yoav Goldberg, Francesco Sartorio, and Giorgio Satta. A tabular method for dynamic oracles in transition-based parsing. *TACL*, 2:119–130, 2014.

[12] Carlos Gómez-Rodríguez and Daniel Fernández-González. An efficient dynamic oracle for unrestricted non-projective parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 256–261, 2015.

[13] Carlos Gómez-Rodríguez, Francesco Sartorio, and Giorgio Satta. A polynomial-time dynamic oracle for non-projective dependency parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–927, Doha, Qatar, October 2014. Association for Computational Linguistics.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.

[15] Jens Nilsson and Joakim Nivre. Malteval: an evaluation and visualization tool for dependency parsing. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008. European Language Resources Association (ELRA).

[16] Joakim Nivre. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 149–160, 2003.

[17] Joakim Nivre. Algorithms for deterministic incremental dependency parsing. *Comput. Linguist.*, 34(4):513–553, December 2008. doi: 10.1162/coli.07-056-R1-07-027.

[18] Joakim Nivre. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 351–359, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[19] Joakim Nivre, Johan Hall, and Jens Nilsson. Maltparser: A data-driven parser-generator for dependency parsing. In *In Proceedings of LREC*, 2006. Available at `http://www.maltparser.org`.

[20] Joakim Nivre, Marco Kuhlmann, and Johan Hall. An improved oracle for dependency parsing with online reordering. In *Proceedings of the 11th International Conference on Parsing Technologies*, IWPT '09, pages 73–76, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[21] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

[22] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951. doi: 10.1214/aoms/1177729586.

[23] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, pages 661–668, 2010.

[24] Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, pages 627–635, 2011.

[25] Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 455–465, 2013.

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[27] Univeral Dependencies Treebanks version 1.2, Released Nov 15, 2015 2015. Permanent identifier for download `http://hdl.handle.net/11234/1-1548`, documentation at `http://universaldependencies.github.io/docs/`.

[28] Hiroyasu Yamada and Yuji Matsumoto. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*, volume 3, pages 195–206, 2003.

[29] Matthew D. Zeiler. Adadelta: An adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[30] Daniel Zeman. Reusable tagset conversion using tagset drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008. European Language Resources Association (ELRA).

[31] Yue Zhang and Joakim Nivre. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 188–193, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

Deploying the New Valency Dictionary Walenty in a DCG Parser of Polish

Marcin Woliński

Institute of Computer Science
Polish Academy of Sciences
E-mail: wolinski@ipipan.waw.pl

Abstract

This paper reports on developments in the Świgra parser related to the availability of the valency dictionary Walenty and their influence on the Składnica treebank of Polish. A method is proposed, which allows to use the rich valency data and yet to avoid unnecessary re-computation and reduplication of syntactic structures.[1]

1 Walenty – a valency dictionary of Polish

Walenty (Hajnicz et al., 2015; Przepiórkowski et al., 2014c,b,a) is a new comprehensive valency dictionary of Polish based on corpus data. Development of Walenty started by enhancing the dictionary created for Świgra parser (see below), but now the dictionary is much larger than the original and provides much richer information. In particular, the new dictionary includes not only verbs but also nouns, adjectives and adverbs. Walenty describes coordination of syntactically different arguments within a single syntactic position (so called unlike coordination), uses structural case (including partitive), provides semantic classification of some adverbial-like arguments (e.g., ablative and adlative), describes control and raising, and includes a rich phraseological component. Moreover, its syntactic level is being currently complemented with semantic frames.[2]

The following example depicts the syntactic schema of the verb CHCIEĆ 'want' used in the tree of Figure 1:

```
subj,controller{np(str)}
+controllee{np(str);cp(żeby);infp(_);advp(misc)}
```

[1] Work financed as part of the investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education.

[2] The semantic level of Walenty is not yet used in Świgra, but it is a planned extension.

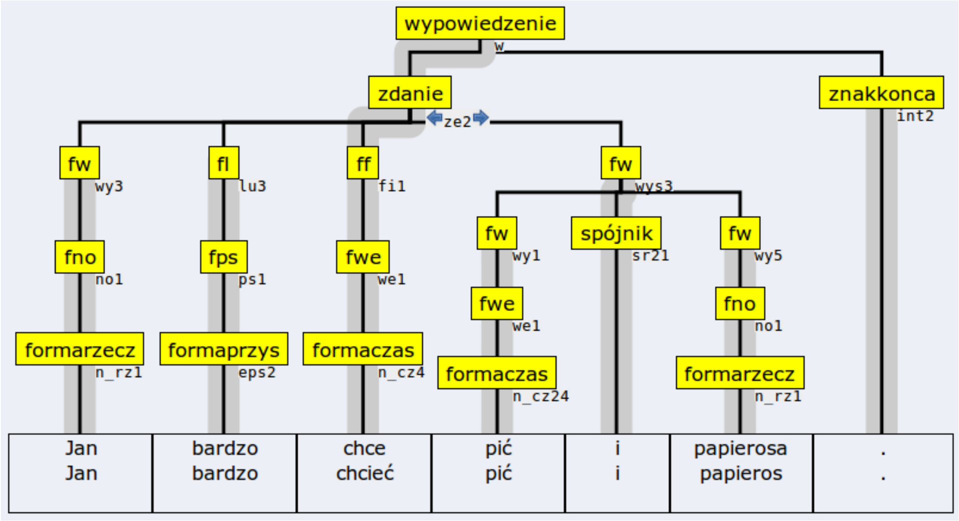


Figure 1: Parse tree for sentence (1)

According to Walenty a verb opens several syntactic positions which can be filled with specific arguments. The example schema comprises two syntactic positions (separated with a +). The first is marked as a subject realised by a nominal phrase in structural case `np(str)`. The second position specifies a list of argument types: a nominal phrase `np` in structural (in this case accusative or genitive depending on negation), a clause `cp` introduced by the complementizer `żeby`, an infinitival phrase `infp` in any aspect, an adverbial phrase `advp` of type `misc`. This notation means that the position can be filled by any of the listed arguments or by a coordination thereof.

Two positions are specially labelled: subject `subj` (the argument in this position influences morphological features of the verb) and passivable object `obj` (the argument in this position turns into a subject in passive voice). Other positions are unlabelled.

The two positions in the example are linked with a control relation expressed with the tags `controller` and `controllee`. By convention, control relations in Walenty are marked against positions, but it is understood that only some argument types take part in these relations. In this case the relation will hold between the argument filling the subject position and the subordinate clause or the infinitival complement.

In Walenty, due to the free word order of Polish, the order of positions within a schema and the order of argument types within a position is not important.

2 The treebank Składnica and the parser Świgra

Składnica (Woliński et al., 2011) is a treebank of Polish built on a 20,000 sentence subcorpus sampled from the manually annotated part of the National Corpus of Polish (Przepiórkowski et al., 2011). The primary format are constituency trees generated with the DCG (Pereira & Warren, 1980) parser Świgra (Woliński, 2004; Świdziński & Woliński, 2010) and then manually disambiguated and validated. The grammar stems from Świdziński's grammar (1992). Currently the treebank contains validated structures for 10,673 sentences.

Figure 1 shows Składnica-style annotation for the sentence:

(1) *Jan bardzo chce pić i papierosa.*
John much want to drink and cigarette

'John wants to drink and a cigarette very much.'

For terminals in the tree, the form and the lemma are shown. Internal nodes are represented by the name of the non-terminal category. But in fact each node carries several attributes specifying its syntactic features. (The attributes can be examined in the interactive interface of the parser.) One of these attributes is the type of an argument, as specified by Walenty.

In the example, the node for sentence (`zdanie`) consists of a 'required phrase' (`fw`, argument); a 'free phrase' (`fl`, adjunct); finite phrase (`ff`); and another `fw`. This last argument is a phrase featuring unlike coordination where a verbal phrase `fwe` in infinitive got coordinated with a nominal phrase `fno` in accusative, as allowed by the Walenty entry quoted in the previous section.

3 Deploying Walenty

3.1 Representation of valency schemata

Valency schemata given by Walenty are maximal in the sense that the dictionary does not list possible sub-schemata of a given schema. In Polish most of arguments are optional (in particular subjects are often omitted; see Section 3.3 for the full list of obligatory arguments in Walenty). Thus a method is needed to allow for the schemata to be realised partially in a controlled way.

One possible solution, used in the LFG grammar POLFIE (Patejuk, 2015), which also uses Walenty, is to compute all subsets of schemata in advance. Each subset leads to a separate lexical entry for the verb. This has the disadvantage of multiplying the lexical entries exponentially: a schema of length n has 2^n subsets (including the empty one).

Schema lengths in Walenty are listed in Table 1. The median of lengths in the dictionary is 3, which means a typical schema gets rewritten into 8 lexicon entries. Moreover, verbs usually have several schemata in Walenty, which leads to the average of 33 lexical entries per verb (even taking into account several schemata having the same sub-schema, e.g., counting a singleton subject as one entry). Maximal number of lexical entries generated this way is 813 for the verb *dać* 'to give'. To make things worse, frequent verbs have more complicated valency than less frequent ones. If we take into the account frequencies of verbs we arrive at the average number of POLFIE style lexical entries equal 76 (counted on the Składnica corpus).

The solution used in POLFIE seems to be motivated by the limitations of LFG (or its implementation XLE), namely by the need to pass the valency information through lexicon entries. We have decided to take a different route. In DCG we have the advantage of being able to program arbitrary conditions, as if extending the formalism for the needs of a particular grammar. In particular, we can manipulate complex valency information during parsing.

We have decided to represent valency information in a form close to the source form of Walenty: a complete list of schemata for the given verb is passed to the parser (both reflexive and non-reflexive readings). Each schema is a list of syntactic positions. Each position is a list of argument specifications.

3.2 Filling syntactic positions

When the parser builds a node for a finite sentence it collects dependents for the given verb or rather for a verbal phrase with this verb as the centre. (We use the finite sentence as an example here, but the same type of processing occurs at all places when arguments are expected by some entity, be it a verb, a noun, an adjective or an adverb). The algorithm maintains two lists: a list of already recognised arguments and a structure representing arguments that can still be added to the interpretation being constructed. The first list is initialised as empty, the second – with the complete valency entry for the verb.

When a new candidate for an argument is considered the following operations need to be performed:

1. Find the set of all schemata that contain positions that contain the type of the given argument.

2. From all of these schemata remove the position containing the argument in question. Note that positions are understood as alternatives: when one argument realising a position is recognised, the whole position is removed as already realised. The result becomes the new list of not yet realised arguments.

3. Add the current argument to the list of already recognised arguments.

These steps are repeatedly applied to all arguments of the verb found in a given sentence.

length	1	2	3	4	5	6	7	8
no. of schemata	282	10701	29048	14419	2897	427	77	3

Table 1: Schema lengths in Walenty

3.3 Argument specifications

As said above, syntactic positions are sets (technically: lists) of argument specifications. These specifications again have some internal structure.

First of all to ease the processing we have decided to represent explicitly the information whether a given argument is obligatory (`obg`) or optional (`opt`).

In Walenty all arguments are optional with the following exceptions:

- All lexicalised (phraseological) arguments are obligatory.[3]
- An argument marked as `controller` is obligatory if its `controllee` is present.

In Świgra we treat the reflexive marker *się* as a special type of argument. This argument is obligatory in finite uses of verbs, but when a schema is derived for a gerund, the argument becomes optional. It is skipped completely when a schema for past participle is derived.

When the parsing algorithm finishes processing of arguments, the list of not yet used parts of schemata is checked against obligatory arguments. All schemata containing unrealised obligatory arguments are deleted from the list. The interpretation is accepted if the resulting list of schemata is nonempty, which means there was at least one schema whose all obligatory arguments were realised.

The second, most obvious, element of argument specification is its type, represented exactly as in the source dictionary.

The third part can contain additional information that further restricts the arguments. For example, a canonical subject is represented by the triple

```
opt/np(str)/subj(G,N,P)
```

where `G`, `N`, and `P` are Prolog variables unified by the algorithm with the values of gender, number, and person of the verb. When a given nominal phrase is to become a subject, its values of the respective categories are required to unify so that an agreement is maintained. A similar mechanism is used to enforce agreements between arguments resulting from the control relations described in Walenty.

3.4 Arguments coordinated within a position

To allow for unlike coordination rules were added to the grammar that allow required phrases `fw` to form coordinated structures. An example can be seen in Fig. 1, where required phrases `fw` for *pić* 'to drink' and *papierosa* 'a cigarette' get coordinated with the conjunction *i* 'and' and form a complex required phrase. The resulting required phrase has as its type a list of types of phrases that got coordinated. When matching such an argument against a syntactic position the algorithm checks whether all types in the list are allowed for the given position.

[3] In Świgra we do not yet use schemata containing lexicalised arguments, since for that the grammar itself will have to undergo some form of lexicalisation.

3.5 Example analysis

As an example let us consider the analysis of the following sentence:

(2) *Jan chce, żeby dać mu spokój.*
John want that give him peace

'John wants to be left alone.'

For brevity we list only a few of schemata for the verb CHCIEĆ and we skip control requirements and the obligatory/optional marker. The schemata in Świgra notation take the following form:

```
[ % schema 1
  [[sie], [np(dat)], [infp(_)]],
  % schema 2
  [[np(str)/subj(G,N,P)],
   [np(str), cp(żeby), infp(_), advp(misc)]],
  % schema 3
  [[np(str)/subj(G,N,P)],
   [np(gen), cp(żeby), ncp(gen,żeby)],
   [prepnp(od,gen)]]
]
```

When parsing example (2) the first argument encountered by the parser (working from the left to the right) is the nominal subject *Jan* of type `np(str)`. Since its morphological features agree with that of the verb we can accept this argument. This will result in filtering out schema 1, since it does not contain a subject. Then the subject position will be removed from schema 2 and 3 resulting in:

```
[% schema 2:
  [[np(str), cp(żeby), infp(_), advp(misc)]],
 % schema 3:
  [[np(gen), cp(żeby), ncp(gen,żeby)],
   [prepnp(od,gen)]]
]
```

The second argument is a clause , *żeby dać mu spokój* headed with the complementizer *żeby*. Its type `cp(żeby)` appears in both available schemata. After this step the list becomes:

```
[% schema 2:
  [],
 % schema 3:
  [[prepnp(od,gen)]]
]
```

To finish up we have to check whether any obligatory arguments remain unrealised, but that is not the case. The only obligatory argument was the reflexive marker `sie`. Both schema 2 and 3 allow to finish analysis at this stage.

It is worth noting that the recognised set of arguments can be an instance of schema 2 as well as schema 3. We do not differentiate between them and so only one parse tree gets generated.

4 Some experimental results

Świgra with Walenty dictionary and the adapted grammar was used to parse anew the whole Składnica corpus (20,000 sentences). This version was able to accept 14,103 sentences (70.5%), while the version with the old dictionary accepted 13,194 (66%). Unfortunately, these newly accepted sentences have not yet been validated by the annotators, so we cannot claim that all new structures are correct.

We have checked the structures generated using Walenty against 10,673 already accepted trees of Składnica. The tree previously accepted by the annotators was found among new parses in 10193 cases (95.5%). For the remaining 480 sentences (4.5%) the parser using Walenty did not produce a compatible tree (in 255 cases (2.4%) the new parse forest was empty). These cases will have to be studied carefully, since they show several problems including errors both in Składnica and in Walenty. For some verbs the two dictionaries differ in opinion whether a given dependent should be considered a complement or an adjunct, so these cases will require further discussion.

Since unlike coordination is one of the more advertised features of Walenty, we have also made a preliminary attempt to estimate the frequency of arguments being coordinated in that manner. The rules for coordination within positions were used in 141 sentences of 14103 sentences that were accepted by the parser. We have checked manually all these sentences and found that only 4 are real examples of this type of coordination, which amounts to 0.03% of sentences. This result can be biased by sentences rejected by the parser, but it seems to be in contrast with the claim of Patejuk & Przepiórkowski (2014) that "such coordination of unlike categories is relatively common in Polish."

5 Conclusions

Parsing Polish is to much extent valency driven. Valency schemata for Polish are numerous and complicated. Polish has free word order allowing to shuffle the schemata arbitrarily. Moreover, most of arguments of a verb are optional. These facts pose specific problems in parsing.

In the paper we have shown that with respect to these problems the DCG formalism provides tools leading to a more effective solution than LFG. One problem of this solution is that it has a "procedural" and not purely "constraint based" flavour. We think of it in terms of "when the parser recognises a candidate argument…", “a position is removed from the schema…”, etc. It seems that to express a similar solution in a constraint based formalism like LFG of HPSG some extensions would be needed in these formalisms.

We hope that this humble contribution will provide some food for thought on desirable features of a formalism well suited for parsing languages typologically similar to Polish.

References

Hajnicz, E., Nitoń, B., Patejuk, A., Przepiórkowski, A., & Woliński, M. (2015). Internetowy słownik walencyjny języka polskiego oparty na danych korpusowych. *Prace Filologiczne*, *LXV*, (to appear).

Patejuk, A. (2015). *Unlike coordination in Polish: an LFG account*. Ph.D. dissertation, Institute of Polish Language, Polish Academy of Sciences, Kraków.

Patejuk, A., & Przepiórkowski, A. (2014). Synergistic development of grammatical resources: a valence dictionary, an LFG grammar, and an LFG structure bank for Polish. In V. Henrich, E. Hinrichs, D. de Kok, P. Osenova, & A. Przepiórkowski (Eds.) *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT 13)*, (pp. 113–126). Tübingen, Germany: Department of Linguistics (SfS), University of Tübingen.
URL `http://tlt13.sfs.uni-tuebingen.de/tlt13-proceedings.pdf`

Pereira, F., & Warren, D. H. D. (1980). Definite clause grammars for language analysis–a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, *13*, 231–278.

Przepiórkowski, A., Bańko, M., Górski, R. L., & Lewandowska-Tomaszczyk, B. (Eds.) (2011). *Narodowy Korpus Języka Polskiego*. Warsaw: Wydawnictwo Naukowe PWN.

Przepiórkowski, A., Hajnicz, E., Patejuk, A., & Woliński, M. (2014a). Extended phraseological information in a valence dictionary for NLP applications. In *Proceedings of the Workshop on Lexical and Grammatical Resources for Language Processing (LG-LP 2014)*, (pp. 83–91). Dublin, Ireland: Association for Computational Linguistics and Dublin City University.
URL `http://www.aclweb.org/anthology/siglex.html#2014_0`

Przepiórkowski, A., Hajnicz, E., Patejuk, A., Woliński, M., Skwarski, F., & Świdziński, M. (2014b). Walenty: Towards a comprehensive valence dictionary of Polish. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.) *Proceedings of the Ninth*

International Conference on Language Resources and Evaluation, LREC 2014, (pp. 2785–2792). Reykjavík, Iceland: ELRA.
URL `http://www.lrec-conf.org/proceedings/lrec2014/index.html`

Przepiórkowski, A., Skwarski, F., Hajnicz, E., Patejuk, A., Świdziński, M., & Woliński, M. (2014c). Modelowanie własności składniowych czasowników w nowym słowniku walencyjnym języka polskiego. *Polonica*, *XXXIII*, 159–178.

Świdziński, M. (1992). *Gramatyka formalna języka polskiego*. Rozprawy Uniwersytetu Warszawskiego. Warszawa: Wydawnictwa Uniwersytetu Warszawskiego.

Świdziński, M., & Woliński, M. (2010). Towards a bank of constituent parse trees for Polish. In P. Sojka (Ed.) *Text, Speech and Dialogue, 13th International Conference, TSD 2010, Brno, September 2010, Proceedings*, vol. 6231 of *LNAI*, (pp. 197–204). Heidelberg: Springer.

Woliński, M. (2004). *Komputerowa weryfikacja gramatyki Świdzińskiego*. Ph.D. thesis, Instytut Podstaw Informatyki PAN, Warszawa.

Woliński, M., Głowińska, K., & Świdziński, M. (2011). A preliminary version of Składnica—a treebank of Polish. In Z. Vetulani (Ed.) *Proceedings of the 5th Language & Technology Conference*, (pp. 299–303). Poznań.

Part II

Short Papers

Automatic Conversion of the Basque Dependency Treebank to Universal Dependencies

Maria Jesus Aranzabe, Aitziber Atutxa, Kepa Bengoetxea, Arantza Diaz de Ilarraza, Iakes Goenaga, Koldo Gojenola and Larraitz Uria

IXA NLP Research Group
University of the Basque Country (UPV/EHU)
E-mail: koldo.gojenola@ehu.eus

Abstract

This work describes the process of automatically converting the Basque Dependency Treebank to Universal Dependencies (UD). Our objective is to develop a set of conversion rules that will automatically transform the original treebank to UD. Basque is a morphologically rich and agglutinative language, which presents different challenges for the conversion from the initial annotation scheme to UD. We will illustrate the steps pursued and the main difficulties we have encountered. As a main conclusion we can say that, although the Basque original treebank was in accord with many UD guidelines, the process was not trivial, converting around 80% of the tokens.

1 Introduction

In this work we describe the conversion of the Basque Dependency Treebank (BDT) to Universal Dependencies (UD) [1, 2, 3, 4]. Although the Basque original treebank was in accord with many UD guidelines, the conversion process presents different challenges. We will try to give a general overview of the process but we will also concentrate on the phenomena where we found some difficulties, specially ellipsis, copulative sentences or multiword units. The Basque language can be described as a morphologically rich, agglutinative language with a high capacity of generating inflected word-forms, with free constituent order of sentence elements. It can be considered a head-final language, as the syntactic head of phrases is located at the end of the last word of the phrase, in the form of a suffix. BDT [5] is a pure dependency treebank from its original design, annotated in the CoNLL-X format, and it shares with UD a lexicalist hypothesis in syntax, where dependencies occur between whole individual wordforms. Under this lexicalist approach, each word shows several morphosyntactic associated features, corresponding to affixes (prefixes and suffixes) attached to the base forms, such as case (there are 14 morphological cases in Basque), number, definiteness or type of subordinate sentence

Table 1: Mapping between BDT and UD for POS tags and dependency relations

Type of mapping (BDT → UD)	**POS tags**	**Dependencies**
1:1	ADJ, ADB, . . . (13 categories)	15 dependencies
1:2	Det → DET/NUM Noun → DET/NUM/PROPN	cmod → advcl/acl detmod → det/nummod
1:5		ncmod → advmod/amod/det/nmod/neg

(adversative, conditional, ...). These suffixes usually appear as separated word-forms in non agglutinative languages. The last version of BDT contains 150,000 tokens forming 11,225 sentences, with 1.3% of non-projective arcs. BDT encodes 16 different POS and 28 dependencies, an extended inventory based on [6].

2 Description of the Automatic Conversion Process

UD covers three levels of annotation: part of speech (POS) [3], morphosyntactic features [4] and dependency labels [1]. The first step of the conversion process consisted of analyzing BDT and UD[1] guidelines in order to find the correct mapping of each Basque tag or dependency label. Mapping POS and morphosyntactic features was a quite straightforward step, described in subsection 2.1. Regarding the conversion of dependencies, there are several phenomena that are worth mentioning, which are presented in the following subsections.

2.1 Conversion of POS and Morphosyntactic Features

Table 1 presents the main differences between the set of POS tags used in BDT and those in UD. The table shows, in its second column, that several of the BDT POS tags have a unique correspondence in UD. However, there are different cases where the mapping is not direct, because a part of speech tag must be mapped to several UD POS tags, depending on other aspects, such as morphological features. This happens with determiners and nouns. On the other hand, there are cases when two different BDT tags are mapped to the same UD POS tag, as in the case of main verbs, which in UD have a unique category (VERB), while there are two tags for Basque main verbs, depending on whether the verb must be accompanied by an auxiliary or it is a *compact* verb where the main verb contains inflectional suffixes corresponding to the auxiliary. This distinction is missed in the UD POS tag, although it can be recovered from the morphosyntactic tags.

Regarding the set of morphosyntactic features, it can be considered the easiest step, as the inventory of UD features was compiled over a big set of dependency treebanks and annotation guidelines [4]. The main differences can be related to differences of specificity, either from BDT or UD, where one of the descriptions

[1] http://universaldependencies.github.io/docs/

gives a more ample set of values for a given category (e.g., the UD guidelines present a wider spectrum of values for numerals, compared to BDT).

2.2 Conversion of Dependencies

Table 1 shows in its third column that, although most of the dependencies are mapped in a straightforward manner, some other are more complex, as in the case of the non-clausal modifier (ncmod) relation in BDT, which is mapped to 5 different relations in UD. Apart from this fact, there have been some other aspects that are presented in the following paragraphs.

Morphological ellipsis

Basque allows the formation of ellipsis inside a wordform, by means of a subordinated relative clause or a genitive, as in

dakarrena (the one that (he) brings) = *dakarren* (that brings (he)) + *-a* (the one)

This wordform presents an example of a relative clause that, when combined with a definite article, forms an ellipsis. As the wordform must be assigned a unique part of speech, it could correspond to either a verb from its original root or a noun, taking its function into account (the whole word acts as an object). Figure 1 shows an example of a sentence that illustrates this phenomenon. The figure shows that this word depends on the main verb by means of a *dobj* relation, which seems contradictory since the word is marked as a verb. Figure 2 shows a sentence parallel to that of Figure 1, but without the ellipsis. In the example, the wordform *Gizonak* (the man) acts as a subject of the subordinated verb, which in turn modifies *gauza* (the thing) by a relative clause (relcl) dependency, and this will be the direct object of the main verb.

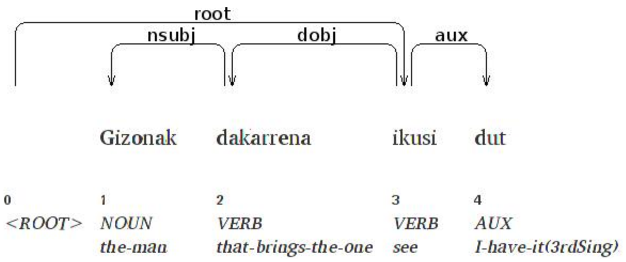


Figure 1: Example of an elliptical relative sentence inside a nominal wordform (*I have seen the one that the man brings*).

Universal Dependency annotation follows a lexicalist view of syntax, which means that dependency relations hold between *words* as in figure 1. Under this view the parallelism that should hold between figure 1 and figure 2 disappears. Universal Dependencies allow some exceptions to the lexicalist view such as Spanish clitics. Up to present, there is agreement on the fact that the lexicalist view should

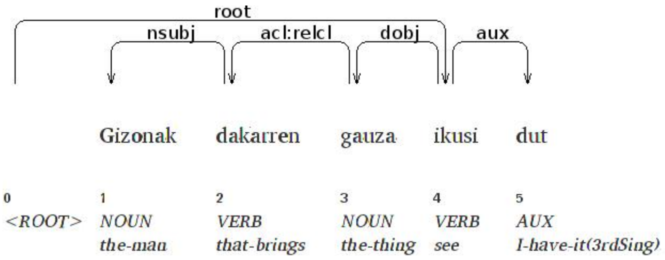


Figure 2: Example of a non-elliptical sentence parallel to the one in Figure 1 (*I have seen the thing that the man brings*).

be followed avoiding splitting as much as possible. Figure 3 presents a possible solution to the problem showed in Figure 1, by separating the verbal and nominal information inside the wordform *dakarrena*. This way, the analysis in the figure is symmetric to that in figure 2, and the verb/noun dichotomy present in figure 1 is solved. Although Basque presents a high rate of morphological ambiguity, we think that the splitting could be done automatically.

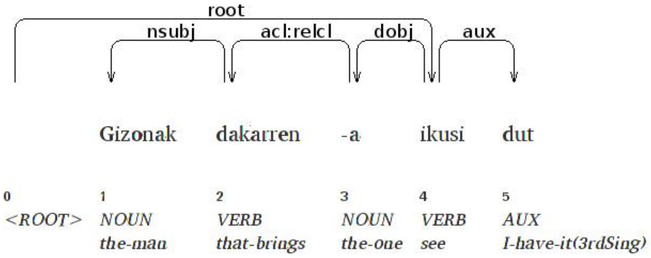


Figure 3: Alternative analysis of the sentence in Figure 1

Multiwords (MWs)

The BDT guidelines allow to agglutinate several wordforms in MWs. Although there are many different combinations creating multiwords, we only transformed the most frequent combinations of POS and CPOS (coarse POS) tags, accounting for 2/3 of the total number of MWs, and leaving the rest for future work. The transformation consists of recovering the original wordforms with their corresponding POS, CPOS and features, assigning at the same time the dependency. An aspect that deserved a careful study was to determine the head and the dependent(s) of each MW. This was easy for compounds, but more difficult with NEs and complex postpositions, as in some of them the words can be inflected, giving different options for choosing the head and dependent. There are three types of MWs in BDT:

- Compounds.
- Named entities (NE), including person, location, organization and undefined (for other types of NEs). These MWs present different patterns for the con-

version, as there are a variety of types of elements, such as nouns, adjectives, adverbs, and numerals.

- Complex postpositions like *mendiaren gainean*:

  mendiaren gainean (on top of the mountain) = *mendiaren* (of the mountain) + *gainean* (on top)

  In this example, both wordforms are inflected with the genitive case and the innessive case, respectively. Although at first sight it could be stated that *mountain* could be the head of the MW unit, the genitive acts as a complement and suggests that *top* is the head.

Coordination

There are several ways of coding coordinated structures, depending on the head of the coordination structure. In BDT the conjunction is the head, while in UD the first argument of the conjuctions acts as the head of the whole structure. Allowing the conjunction to be the head of the coordination as in BDT can better represent certain scope phenomena and ellipsis occurring through coordination, because the UD specification for coordination, attaching all the elements to the first conjunct, loses some scope information present in the original BDT such as, for example, in figure 4, when a modifier is a dependent of the whole coordinated sequence.

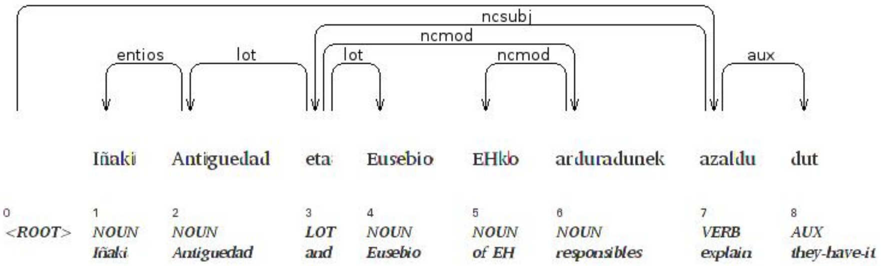


Figure 4: Analysis in BDT where the conjunction *eta* is the head and the scope of the modifier (*the responsibles of EH* linked by a *ncmod* dependency relation) applies over the whole coordination structure (*Iñaki Antiguedad and Eusebio the responsibles of EH have explained (it)*).

In addition, allowing the conjunction to be the head of the coordination favours representing coordinative ellipsis as, for example in figure 5, where two sentences are linked by a coordination conjunction (*eta*), and the second sentence does not contain a main verb (ellipsis). As shown in figure 6, the parallelism occurring in coordinate ellipsis did not get captured after the converstion to UD. One way of solving it could be to add some especificity over the *conj* relation for capturing the symmetry, as presented in figure 7. No decision has been taken in the UD community so far, and coordinate ellipsis remains problematic. In fact, figure 6 is the actual conversion for the original BDT sentence (see figure 5).

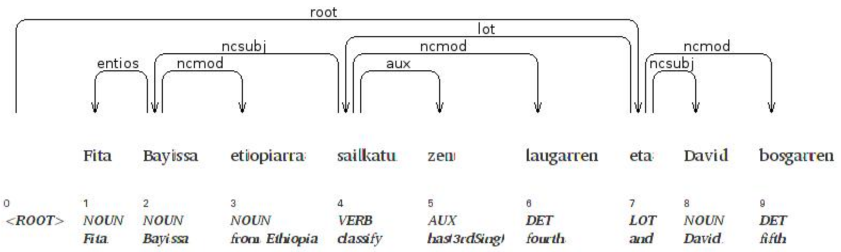


Figure 5: Analysis in the BDT where the conjunction is the head *and* and acts as a place holder for ellipsis (*Fita Bayissa from Ethiopia has classified fourth and David fifth*).

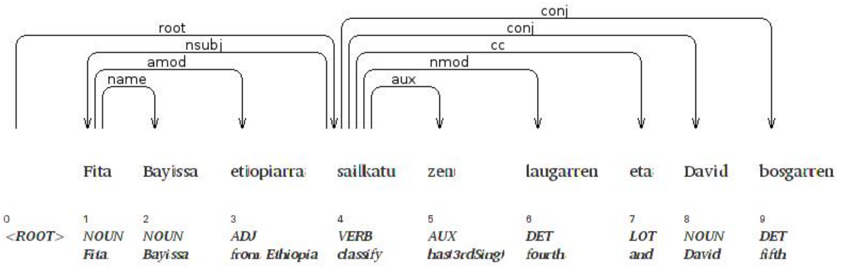


Figure 6: Analysis after the UD conversion where the first conjunct is the head of the coordination.

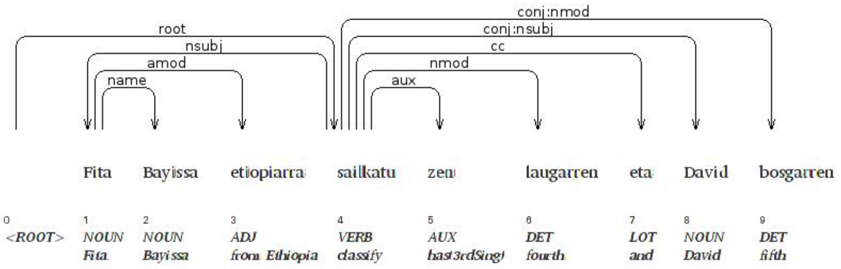


Figure 7: Alternative analysis after the UD conversion where the first conjunct is the head of the coordination.

Copulative sentences

Although the UD guidelines only allow copulative sentences using *be* as the copula (this restriction is an open issue in the UD community), in Basque several verbs can take part in these sentences, and they need additional analysis. Usually there is agreement between the copulative modifier and the subject, whereas with predicative verbs the modifier is adverbial and does not show agreement.

3 Results

The above presented criteria were transformed in a set of scripts for the automatic conversion from BDT to UD. The order of transformations is not trivial, since changing a part of the treebank can have consequences on subsequent conversions. For example, converting some dependencies needs an examination of the original BDT tags, and for this reason we had to maintain both the original tags together with the UD tags. Generally, more abstract conversions should be applied first, such as the transformation of coordinated sentences, because changing lower level constructions could give erroneous results.

For each phenomena mentioned in subsection 2.2 we first performed a quantitative and qualitative study, and oriented our study towards the design of a set of rules dealing with the most frequent patterns, giving priority to coverage, but without compromising precision, that is, we did not convert any instance not covered in the patterns. This process will leave out a subset of sentences of each phenomena, which are left as future work. A potential side effect will be that some low-frequency phenomena will not be covered by the UD treebank.

As a result of the previously described conversion steps, we have obtained a UD based Basque treebank containing 121,000 tokens, which represents around 80% of the sentences in the BDT. On one hand, this can be seen as a succesful accomplishment, since the conversion rules were designed taking a conservative approach, with the aim of achieving high precision and not leaving any room for conversion errors. On the other hand, the set of remaining sentences correspond to either special cases not accounted by the conversion rules or other types of less frequent phenomena which have not been dealt with at the moment.

4 Conclusion

Although the annotation of the Basque Dependency Treebank (BDT) is in accord with most of the UD guidelines (for example, taking content words as heads), the conversion has been a complex task, from the relatively direct mappings of POS tags to more complex phenomena like ellipsis, copulative sentences or multiwords. At the moment, a set of sentences (120,000 tokens) has been successfully converted, but there are some issues that need to be addressed in order to convert the remaining part of BDT.

Overall, we can state that, except for several phenomena where we have found some difficulty, the automatic conversion process is feasible. We can also say that some of the problematic issues are shared in several cases with typologically similar languages like Finnish or Turkish, and in this respect they can serve to adapt the UD guidelines in order to generalize over the whole set of languages involved.

Acknowledgments

This research was supported by the the Basque Government (IT344-10, DETEAMI), the University of the Basque Country (INF13/59) and the Spanish Ministry of Science and Innovation (MINECO TIN2013-46616-C2-1-R).

References

[1] de Marneffe, Marie-Catherine, Dozat, Timothy, Silveira, Natalia, Haverinen, Katri, Ginter, Filip, Nivre, Joakim, Manning, Christopher D. (2014) *Universal Stanford dependencies: A cross-linguistic typology. Proceedings of the Language Resources and Evaluation Conference (LREC14).* Reykjavik, Iceland

[2] Mcdonald, Ryan, Nivre, Joakim, Quirmbach-brundage, Yvonne, Goldberg, Yoav, Das, Dipanjan, Ganchev, Kuzman, Hall, Keith, Petrov, Slav, Zhang, Hao, Täckström, Oscar, Bedini, Claudia, Bertomeu Castelló, Núria, Lee, Jungmee (2013) *Universal Dependency Annotation for Multilingual Parsing*, In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 92–97, Sofia, Bulgaria

[3] Petrov, Slav, Das, Dipanjan, McDonald, Ryan (2012) *A Universal Part-of-Speech Tagset* In Calzolari, Nicoletta, Choukri, Khalid, Declerck, Thierry, Uğur Doğan, Mehmet, Maegaard, Bente, Mariani, Joseph, Moreno, Asuncion, Odijk, Jan, Piperidis, Stelios (eds.) European Language Resources Association (ELRA) *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC12).* Istanbul. Turkey

[4] Zeman, Daniel (2008) *Reusable Tagset Conversion Using Tagset Drivers. Proceedings of the Sixth International Language Resources and Evaluation Conference (LREC08)*, pp. 28–30, Marrakech, Morocco

[5] Aduriz, Itziar, Aranzabe, Maria Jesus, Arriola, Jose Mari, Atutxa, Aitziber, Diaz de Ilarraza, Arantza, Garmendia, Aitzpea, Oronoz, Maite (2003). *Construction of a Basque Dependency Treebank.* Treebanks and Linguistic Theories, pp. 201-204. Vaxjo, Sweden.

[6] Carroll, John, Briscoe, Ted, Sanfilippo, Antonio (1998). *Parser Evaluation: a Survey and a New Proposal*, *Proceedings of the First International*

Conference on Language Resources and Evaluation (LREC98)., pp.474-454. Granada, Spain

[7] Mel'čuk, Igor (1988) *Dependency Syntax: Theory and Practice*. State University of New York Press.

[8] Zeman, Daniel, Mareček, David, Popel, Martin, Ramasamy, Loganathan, Štěpánek, Jan, Žabokrtský, Zdeněk, Hajič, Jan (2012) *HamleDT: To Parse or Not to Parse?* In Calzolari, Nicoletta, Choukri, Khalid, Declerck, Thierry, Uğur Doğan, Mehmet, Maegaard, Bente, Mariani, Joseph, Moreno, Asuncion, Odijk, Jan, Piperidis, Stelios (eds.) *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC12)*, pp. 2735-2741, Istanbul, Turkey

The Dundee Treebank

Maria Barrett, Željko Agić and Anders Søgaard

Center for Language Technology, University of Copenhagen
Njalsgade 140, 2300 Copenhagen S, Denmark
E-mail:{barrett|zeljko.agic|soegaard}@hum.ku.dk

Abstract

We introduce the Dundee Treebank, a Universal Dependencies-style syntactic annotation layer on top of the English side of the Dundee Corpus. As the Dundee Corpus is an important resource for conducting large-scale psycholinguistic research, we aim at facilitating further research in the field by replacing automatic parses with manually assigned syntax. We report on constructing the treebank, performing parsing experiments, as well as replicating a broad-scale psycholinguistic study—now for the first time using manually assigned syntactic dependencies.

1 Introduction

The *Dundee Corpus* is a major resource for studies of linguistic processing through eye movements. It is a famous resource in psycholinguistics, and—to the best of our knowledge—the world's largest eye-movement corpus. The English part of the Dundee Corpus was annotated with part-of-speech (POS) information in 2009 [9]. This layer of annotation facilitated new psycholinguistic studies such as testing several reader models using models of hierarchical phrase structure and sequential structure [10].

In this paper, we describe a recent annotation effort to add a layer of dependency syntax on top of the POS annotation, enabling the replication of classic studies such as [8] on manually assigned syntax rather than automatic parses. We first describe the Dundee Corpus, then our annotation scheme, and finally we discuss applications of this annotation effort.

2 The Dundee Corpus

The Dundee Corpus was developed by Alan Kennedy and Joël Pynte in 2003, and it contains eye movement data on top of English and French text [13]. Measurements were taken while participants read newspaper articles from *The Independent* (English) or *Le Monde* (French). Ten native English-speaking subjects participated

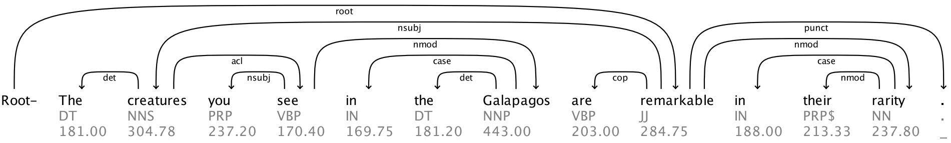


Figure 1: An example sentence (#10) from the Dundee Corpus with UD-style syntactic dependencies and per-word fixation durations.

in the English experiments reading 20 articles, which we focus on here. For a more detailed account, see [14].

The English corpus contains 51,502 tokens[1] and 9,776 types in 2,368 sentences. The apparatus was a Dr Bouis Oculometer Eyetracker with a 1000 Hz monocular (right) sampling. The corpus provides information on fixation durations and fixation order on word level—while also accounting for landing position—for a relatively natural reading scenario. Subjects read running text, 5 lines per display.

Eye movements provide a window to the workings of the brain, e.g. by reflecting cognitive load. Recordings of eye movements during reading is one of the main methods for getting a millisecond to millisecond record of human cognition. Eye movements during reading is controlled by a complex interplay between low-level factors (how much the eye can see and encode from each fixation, word length, landing position, etc.) and high-level factors (e.g. syntactic processing). For an overview, see [18].

This resource has enabled researchers to study things like syntactic and semantic factors in processing difficulty of words [16] and whether the linguistic processing associated with a word can proceed before the word is uniquely identified [19].

3 Syntactic Annotation

In annotating the Dundee Corpus for syntactic dependencies, we follow the Universal Dependencies (UD) guidelines[2] [1] as the emerging *de facto* standard for dependency annotation.

The guidelines build on—and closely adhere to—Universal Stanford dependencies [7], proposing 40 dependency relations together with an universal POS tagset (UPOS) and morphological features. We convert the Penn Treebank-style POS tags from the Dundee Corpus into UPOS, and we provide the universal morphology features, by using the official English UD conversion tools.

The guidelines for annotating English are very well-documented within the UD framework. We only briefly touch upon the most important ones.

For core dependents of clausal predicates, UD distinguishes between nominal subjects (NSUBJ), nominal subjects of passives (NSUBJPASS), direct objects

[1] According to the tokenisation of the Dundee corpus where punctuation and contracted words are glued to the preceding word.

[2] `http://universaldependencies.github.io/`

	Dundee			English UD dev			English UD test		
Training set	*LAS*	*UAS*	*LA*	*LAS*	*UAS*	*LA*	*LAS*	*UAS*	*LA*
Dundee	82.23*	85.06*	89.97*	69.50	75.96	81.26	68.86	75.60	80.61
English UD	71.45	78.66	84.28	85.51	88.03	92.91	84.72	87.30	92.37

Table 1: Dependency parsing results with English UD and Dundee as training sets. Parser: `mate-tools` graph-based parser with default settings [5]. Features: FORM and CPOSTAG only, using the Penn Treebank POS tags. Metrics: labeled and unlabeled attachment scores (LAS, UAS), and label assignment (LA). *: 5-fold 80:20 cross-validation, as the Dundee Treebank has no held-out test set.

(DOBJ), indirect objects (IOBJ), clausal subjects (CSUBJ), clausal subjects of passives (CSUBJPASS), clausal complements (CCOMP), and open clausal complements (XCOMP). When it comes to non-nominal modifiers of nouns, for example, the guidelines distinguishes between adjectival modifiers (AMOD), determiners (DET), and negation (NEG).

We show an example sentence from the treebank in Figure 1. It depicts the UD-style dependency annotation, as well as per-word total fixation durations averaged over ten readers. Some of the typical UD-style conventions—such as content head primacy and no copula heads—are also illustrated.

We used two professional annotators that had previously worked on treebanks following the UD guidelines. The annotators provided double annotations for 118 sentences, with moderately high inter-annotator agreements of 80.82 (LAS), 87.61 (UAS), and 86.63 (LA).

Further, we trained a graph-based dependency parser [5] on English UD training data, and parsed the Dundee Corpus text. We report the results in Table 1. There is a decrease in accuracy moving from English UD to the Dundee Corpus text. We attribute the decrease to the domain shift—English UD stemming from various web sources, while Dundee consists of newswire commentaries in specific—and possibly to the slight cross-dataset inconsistency in POS and dependency annotations. In a separate experiment, we also parse the Dundee Corpus text using 5-fold cross-validation with an 80:20 split, observing accuracies consistent with the English UD experiment. These results are also reported in Table 1.

The cross-dataset decrease in parsing accuracy, even if irrelevant for Dundee-specific experiments, plays into the argument for using gold-standard annotations in psycholinguistic research.

4 Replication of Dependency Locality Theory Experiment

The Dundee Treebank annotated with dependencies has the following affordances. First, it allows for replication of studies such as [8] with manual annotations. Sec-

Predictor	Coef	p	Coef original	p original
INTERCEPT	199.59		128.24	***
WORDLENGTH	-1.25		30.90	***
WORDFREQUENCY	4.43	***	14.50	***
PREVIOUSWORDFIXATED	-33.32	***	-18.05	***
LANDINGPOSITION	-1.23	***	-4.18	***
LAUNCHDISTANCE	1.79	***	-1.91	***
SENTENCEPOSITION	-.09	*	-.12	*
FORWARDTRANSITIONALPROBABILITY	1.51	***	-3.27	***
BACKWARDTRANSITIONALPROBABILITY	-5.87	***	3.96	***
log(DLT)	3.51	**	5.86	*
WORDLENGTH:WORDFREQUENCY	-2.96	***	-4.98	***
WORDLENGTH:LANDINGPOSITION	-.68	***	-1.02	***

Table 2: First pass durations for nouns with non-zero DLT score in the Dundee corpus. Coefficients and their significance level. Same predictors as original noun experiment. * $p < .05$, ** $p < .01$, *** $p < .001$.

ond, gaze features can be used to improve NLP models by enabling joint learning of gaze and syntactic dependencies [2, 3]. Finally, the Dundee Treebank facilitates for researchers to study the reading of very specific syntactic constructions in naturalistic, contextualized text, while controlling for individual variation, and variation specific to the parts of speech or syntactic dependencies involved.

Demberg and Keller(D&K) were the first to test broad-covering theories of sentence processing on large-scale, contextualized text with eye tracking data [8]. They explored two theories of syntactic complexity, namely Dependency Locality Theory (DLT) and Surprisal, and how these correlate with three eye tracking measures while controlling for oculomotor and low-level processing.

DLT [11] estimates the computational resources consumed by the human processor and computes a cost for any discourse referent as well as a cost for every discourse referent between a particular discourse referent and it's head. Thus, DLT needs dependency parsed text to score the complexity of the sentences and Minipar was used to parse the text with a reported 83% accuracy of the DLT score.

In this paper we replicate the parts of their experiments involving DLT, but with manually assigned dependencies instead of automatic parses for calculating DLT. D&K found that DLT score did not have the expected positive effect on reading time of all words. The calculation of DLT only applies for nouns and verbs. They did, however, find that DLT significantly had a positive effect on reading times for nouns and verbs.

We replicate the linear mixed-effects experiment using first pass fixation duration per word for all words and nouns[3]. First pass fixation duration is the duration

[3] The original paper does not contain information about the fixed effects of the model for verbs, why this part of the experiment was not replicated.

of all fixations on specific word from the readers eyes first enter into the region and until the eyes leave the region, given that this region is fixated. This is an measure said to encompass early syntactic and semantic processing as well as lexical access. We use the same low-level predictor variables as the original experiment:

1. word length in characters (WORDLENGTH),
2. log-transformed frequency of target word (WORDFREQUENCY),
3. log-transformed frequency of previous word (PREVIOUSWORDFREQUENCY),
4. forward-transitional probability (FORWARDTRANSITIONALPROBABILITY),
5. backward transitional probability (BACKWARDTRANSITIONALPROBABILITY),
6. word position in sentence (SENTENCEPOSITION),
7. whether the previous word was fixated or not (PREVIOUSWORDFIXATED),
8. launch distance of the fixation in characters (LAUNCHDISTANCE),
9. and fixation landing position (LANDINGPOSITION).

Backward- and forward transitional probabilities are conditional probabilities of a word given the previous / next word, respectively [15]. Along with the word frequencies these two measures are obtained from the British National Corpus (BNC) [6], following the line of D&K. We use KenLM [12] for getting the bigram frequencies and Kneser-Ney smoothing for those bigrams that are not found in the training set. D&K used CMU-Cambridge Language Modeling Toolkit and applied Witten-Bell smoothing. Bigrams respect sentence boundaries.

We clean the data following the described approach by using only fixated words, excluding words that are followed by any kind of punctuation and excluding first and last words of each line. We did, however, not remove words "in a region of 4 or more adjacent words that had not been fixated", since it is unclear what a "region" is (non-fixated words are already removed). This left us with 209,010 datapoints. D&K report to have 200,684 datapoints after cleaning. The difference is probably accounted for by the missing, last cleaning step.

We use R [17] and lme4 [4] to fit a linear mixed-effects model. In the following we use the same fixed and random effects as their models minimised using Akaike Information Criterion (AIC). The authors do not report which significance test they used. We use likelihood ratio tests of the full model with the particular fixed effect against the model without the particular fixed effect.

D&K find that for all words, DLT had a significant, negative effect on first pass fixation duration ($p < .001$), which is a displeasing counter-intuitive result. It means higher DLT score gives a shorter fixation duration. We also find a very small negative effect (-.03) of DLT on first pass fixation duration for all words, but it doesn't reach significance. Following the original experiment, we fit a model for the nouns with non-zero DLT score, encompassing 51,786 data points. The original experiment report having 45,038. In Table 2 we report the coefficients and significance level for all fixed effects of this model as well as the corresponding results of the original experiment. Like the original experiment, we find that the log(DLT) had a significant positive effect on reading time ($p < .01$). These two experiments demonstrate that parser bias did not skew the results substantially.

5 Conclusion

We introduced the Dundee Treebank—a new resource for corpus-based psycholinguistic experiments. The treebank is annotated in compliance with the Universal Dependencies scheme. We presented the design choices together with a batch of dependency parsing experiments.

We also partly replicated a study, which explores how a theory of sentence complexity, DLT, is reflected in reading times. We used manually assigned dependencies instead of parsed dependencies. Like the original experiment, we found both a small negative effect of DLT on all word and a significant positive effect of DLT on reading time for nouns with non-zero DLT score.

The treebank is made publicly available for research purposes.[4]

References

[1] Željko Agić, Maria Jesus Aranzabe, Aitziber Atutxa, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Jan Hajič, Anders Trærup Johannsen, Jenna Kanerva, Juha Kuokkala, Veronika Laippala, Alessandro Lenci, Krister Lindén, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Héctor Alonso Martínez, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Joakim Nivre, Hanna Nurmi, Petya Osenova, Slav Petrov, Jussi Piitulainen, Barbara Plank, Prokopis Prokopidis, Sampo Pyysalo, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Kiril Simov, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. Universal dependencies 1.1, 2015. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

[2] Maria Barrett and Anders Søgaard. Reading behavior predicts syntactic categories. In *CoNLL*, pages 345–249, 2015.

[3] Maria Barrett and Anders Søgaard. Using reading behavior to predict grammatical functions. In *CogACLL*, 2015.

[4] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

[5] Bernd Bohnet. Top accuracy and fast dependency parsing is not a contradiction. In *COLING*, 2010.

[6] British National Corpus Consortium et al. British national corpus version 3, 2007.

[4] `https://bitbucket.org/lowlands/release`

[7] Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal stanford dependencies: A cross-linguistic typology. In *LREC*, pages 4585–4592, 2014.

[8] Vera Demberg and Frank Keller. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109:193–210, 2008.

[9] Stefan L Frank. Surprisal-based comparison between a symbolic and a connectionist model of sentence processing. In *CogSci*, pages 1139–1144, 2009.

[10] Stefan L Frank and Rens Bod. Insensitivity of the human sentence-processing system to hierarchical structure. *Psychological Science*, 22(6):829–834, 2011.

[11] Edward Gibson. The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, pages 95–126, 2000.

[12] Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics, 2011.

[13] Alan Kennedy, Robin Hill, and Joël Pynte. The dundee corpus. In *ECEM*, 2003.

[14] Alan Kennedy and Joël Pynte. Parafoveal-on-foveal effects in normal reading. *Vision research*, 45(2):153–168, 2005.

[15] Scott A McDonald and Richard C Shillcock. Low-level predictive inference in reading: The influence of transitional probabilities on eye movements. *Vision Research*, 43(16):1735–1751, 2003.

[16] Jeff Mitchell, Mirella Lapata, Vera Demberg, and Frank Keller. Syntactic and semantic factors in processing difficulty: An integrated measure. In *ACL*, pages 196–206, 2010.

[17] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[18] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372, 1998.

[19] Nathaniel J Smith and Roger Levy. Fixation durations in first-pass reading reflect uncertainty about word identity. In *CogSci*, 2010.

4-Couv, a Backcover Treebank

Philippe Blache,[1,2] Grégoire Montcheuil,[2,3]
Stéphane Rauzy[1,2] and Marie-Laure Guénot[2,3]

[1]Aix Marseille Université, [2]CNRS, [3]Equipex ORTOLANG
E-mail: `firstname.lastname@lpl-aix.fr`

Abstract

We present in this paper *4-Couv*, a treebanking project aiming at developing a multipurpose treebank for French . The main characteristic of this project is to provide adequate material for both linguistic and psycholinguistic research. The treebank is made of short and self-contained texts, selected from a corpus of backcovers coming from different editors. Such material makes possible classical linguistic research in syntax and discourse, but also offers new perspectives in experimental linguistics: the texts being short and semantically coherent, they perfectly fit with the requirements of eye-tracking or electro-encephalographic recordings. At this stage, *4-Couv* contains 3,500 trees automatically tagged and parsed, and manually corrected. Its format is compatible with other French treebanks. This paper presents the corpus, its annotation and several treebanking tools that have been developed for the different stages of its elaboration: text selection, tagging, parsing and tree edition.

1 Introduction

Treebanks, that still constitute an essential resource in linguistic description as well as natural language processing, are now faced with new uses, in particular in the perspective of experimental linguistics and psycholinguistics. We present in this paper a new treebanking project (at this stage for French), *4-Couv*, aiming at answer the needs of different possible perspectives. Before describing the project, let's underline the fact that only a few truly available treebanks exist for French, mainly the *French Treebank* (*FTB*, Abeillé et al. [1]) and its derivatives, or the French part of the *Universal Dependencies Treebank*[1]. However, only few experiments have been done using these resources in the perspective of studying human language processing. They consist in tracking eye-movement when reading texts in the perspective of evaluating difficulty models (on the basis of the number and the length of the fixations). To this day, most of the studies only take into account the

[1] `https://code.google.com/p/uni-dep-tb/`

morphosyntactic level, such as the works done for English (Demberg and Keller [4]) or for French (Rauzy and Blache [11]), using extracts of *FTB*. In these experiments however, the nature of the texts could constitute an important bias: they are taken from the newspaper *Le Monde*, and consist in articles describing the economic situation 20 years ago. They are then of poor interest for a "normal" reader. This problem can induce an effect of "superficial" reading, leading to an important loss of attention as well as an understanding deficit.

In the perceptive of developing such new uses of treebanks, as well as enriching the amount of available data for French, we have created a new treebank based on short texts, semantically consistent and self-contained, and arousing interest so as to maintain the attention during reading.

The treebank is built from a corpus of "backcovers" called *4-Couv*, answering all these needs. This project is still under development, a first release will be done by the end of end 2015. It consists in a set of texts from various publishers (Pocket, Gallimard) that gave their agreement. We collected first 8,000 texts, among which 500 have been selected, representing 3,500 sentences.

We present in this article the methodology and the tools that have been developed to create *4-Couv*. The first section details the nature of the texts, the characteristics of the annotation scheme and the automatic parsing. The second section outlines the tools used for the selection of texts and the revision of annotations.

2 The corpus, its annotations

2.1 The corpus

Backcovers are small texts, containing between 80-200 tokens for 4-10 sentences, generally short (80% of sentences having at most 30 tokens, and less than 10% are longer than 40 tokens). Texts are generally (a) an extract, (b) the synopsis of the story, (c) the genesis of the book, (d) a comment about the work, or (e) a combination of two or three of this elements. Each of these short texts are semantically autonomous and – a fundamental aspect for our purpose – are supposed to keep the reading interest alive, minimizing attention and comprehension drops.

2.2 Lexical annotations

The annotation of minimals syntactic units is based on the lexicon *MarsaLex*[2] that associates each form with its part of speech and morpho-syntactic features. The segmentation into tokens is maximal in that highly constrained forms are split into distinct lexical units as long as they follow syntactic composition rules. For example, constituents of semi-fixed expressions such as *"il était une fois"* (*once upon a time*) or *"mettre à nu"* (*lay bare*) are split, while other multiword expressions such

[2] *MarsaLex*, hdl:11041/sldr000850

Category	**features**
Adjective	nature, type, gender, number, position
Adverb	nature, type
Connector	nature
Determiner	nature, type, person, gender, number
Interjection	
Noun	nature, type, gender, number, referent type
Punctuation	nature
Preposition	type
Pronoun	nature, type, person, gender, number, case, reflective, postposed
Verb	nature, modality, tense, person, gender, number, auxiliary, pronominal, (im)personal, direct object, indirect complement

Figure 1: Lexical categories and features

as *"d'autant plus"* (*all the more*) or *"tant mieux"* (*even better*) are not, as they do not follow any syntactic composition.

Each lexical category has a specific features set (see figure 1), although many features are common to different categories (typically the gender, number, person). The part-of-speech and feature sets are relatively standard and compatible with most of automatically tagged corpus, and enable to indicate a combination of lexical, morphologic, syntactic and occasionally semantic informations that will have effect on the syntactic construction of upper levels, e.g. the number of a determiner, the subcategorization or the case of a clitic pronoun. We do not have discontinuous lexical constituent, and the tagging is disambiguated (i.e. each element have one part-of-speech, whose sub-categories features could be underspecified when necessary). We do not modify the category of units that change their paradigm (*"une tarte maison"* (*an home[made] pie*), *"il est très zen"* (*he is very zen*)).

2.3 Syntactic annotation

In order to maintain interoperability with the *FTB* (even though it could be not direct and require some processing), the treebank is constituency-based and syntactic relations are represented by means of trees. We apply the following formal constraints:

- No empty category is inserted in the trees (e.g. in the case of an elliptical construction), each node is instantiated by a lexical or a phrase-level unit.
- We distinguish between lexical and phrase level: we keep unary phrases, e.g. *Simone* is the unique constituent of a *NP* in (1).

  (1) *"Simone m'en donne trois."* (*Simone gives me three.*)

Phrase-level constructions					
AdP	adverbial phrase	VPinf	infinitive clause	SENT	sentence
AP	adjectivial phrase	VPpart	participial clause	Srel	relative clause
NP	noun phrase	VN	verbal nucleus	Ssub	subordinate clause
PP	prepositional phrase	VNinf	infinitive VN	Sint	other clause
VP	verbal phrase	VNpart	participial VN		

Syntactic functions					
		indirect complement		predicative complement	
SUJ	subject	A-OBJ	- introduced by *à*	ATS	- of a subject
OBJ	direct object	DE-OBJ	- introduced by *de*	ATO	- of a direct object
MOD	modifier or adjunct	P-OBJ	- other preposition		

Figure 2: Syntactic tagset

- No discontinuous constituent or unbounded dependencies directly encoded, such as in (1) or (2).

  (2) *"Ce film, Paul et moi on a adoré."* (*This movie, Paul and I we really do like.*)

- The phrase-level tagset (see figure 2) is reduced to classical phrases, at the exclusion of other constructions such as coordination (at the difference with the *FTB* and its derivatives).

- The same types of syntactic functions than those introduced for the *FTB* (see figure 2) are used. This annotation is less precise then other annotation frameworks (such as Gendner et al. [5]) where structural and functional informations were given independently.

2.4 Parser

The treebank is generated with the LPL stochastic parser[3] (Rauzy and Blache [10]). The processing flow follows a classical scheme. After tokenization, POS-tagging is done by means of a stochastic HMM tagger using Rabiner [9]. Finally, the stochastic parser generates the possible tree structures and selects the most probable one.

The probabilistic model for the POS tagger was trained with the *GraceLPL* corpus, a version of the *Grace/Multi-tag* corpus (Paroubek and Rajman [8]) that contains 700,000 tokens and which we correct and enrich regularly. In the model the morphosyntactic information is organized into 48 distinct tags (version 2013). On this tagset, the score (F-measure) of the tagger is 0.974.

On its side, the parser has been trained with *FTLPL* treebank (Blache and Rauzy [2]), a version of the *MFT* (Schluter and van Genabith [12]) extracted from

[3] *MarsaTag*, hdl:11041/sldr000841

the *FTB* that contains at the moment 1,500 validated sentences with both constituent structure and syntactic functions (around 26,000 tokens).

3 The *4-Couv* treebanking tools

3.1 Text selector

We have developed a tool helping in the texts selection, in the form of HTML files that comes to genuine autonomous wiki[4]. This strategy to use autonomous HTML files allow to easily distribute the revision work between different experts, without needing to install any particular software (files are working directly in most of web browsers[5]), neither to connect with a central server (that allows off-line revision). Each file containing 10 texts to evaluate, presenting the book description, the text segmented into sentences, and an evaluation form (containing check boxes and drop-down lists, see figure 3). The wiki syntax renders easy to correct errors in the sentence division (each sentence is a row in a one-column table) or separate the different parts of the text (inserting a blank line). Furthermore, it also proposes to associate information to unknown words and edit the metadata fields.

3.2 Revision tools

The correction of the automatic annotations is done in two steps. The first concerns the **morphosyntactic tags** and the second consists in the **revision of the constituents trees** produced by the parser.

The morphosyntactic correction tool (see figure 5) presents one token per line, each line containing the form, and a list of possible tags associated to the form, starting with the proposed one. Selecting a new tag consists in clicking another one from the suggested list.

The syntactic correction tool is a tree editor. Only a few of them already exist such as *WordFreak* (Morton and LaCivita [6]) or *TrED 2.0* (Pajas and Štěpánek [7]). More recently, some "web-based" annotation platforms have also been created, offering an intuitive and fast annotation (*brat* (Stenetorp et al. [13]) and sometimes project management facilities (for example by specifying the roles such as annotator, curator or project manager (*GATE Teamware* (Bontcheva et al. [3]) or *WebAnno* (Yimam et al. [14])). However, most of these tools have been developped for dependency-based treebanks. As our approach is constituency based (requiring therefore to deal with a potentially large number of levels), we had to develop a specific editor, that could run in a single HTML (see figure 6) or be integrated into an annotation platform such as *brat* or *WebAnno*.

[4] We customize a *TiddlyWiki* (`http://classic.tiddlywiki.com/`, version 2.8.1) that supply the autonomous wiki, and use a Perl script to "fill" each file with the information.

[5] Only a small plugin could be required to save the modified files.

4 Conclusion

The purpose of this paper is twofold. First it aims to present a new treebank, not only proposing the classical information of this kind of resource in terms of linguistic annotation, but also answering the specific needs of experimental linguistic, in the perspective of acquiring neuro-physiological data on the basis of short and self-contained text. Secondly it also presents new treebanking tools, helping at the different stages of the process: corpus creation, pre-edition, and manual correction of the automatically generated parses. A first resource of 500 texts (3,500 trees) has been created to be distributed, together with the tools, by the end of 2015.

Acknowledgments

This work has benefit from the support of ORTOLANG (ANR-11-EQPX-0032), BLRI (ANR-11-LABX-0036) and A*MIDEX (ANR-11-IDEX-0001-02).

References

[1] A. Abeillé, L. Clément, and F. Toussenel. Building a treebank for french. In A. Abeillé, editor, *Treebanks*, Kluwer, Dordrecht, 2003.

[2] Philippe Blache and Stéphane Rauzy. Enrichissement du FTB: un treebank hybride constituants/propriétés. In *Actes de TALN*, 2012.

[3] Kalina Bontcheva, Hamish Cunningham, Ian Roberts, Angus Roberts, Valentin Tablan, Niraj Aswani, and Genevieve Gorrell. Gate teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4):1007–1029, 2013. ISSN 1574-020X. doi: 10.1007/s10579-013-9215-6. URL http://dx.doi.org/10.1007/s10579-013-9215-6.

[4] Vera Demberg and Frank Keller. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210, 2008.

[5] Véronique Gendner, Anne Vilnat, Laura Monceaux, Patrick Paroubek, Isabelle Robba, Gil Francopoulo, and Marie-Laure Guénot. Les annotation syntaxiques de référence PEAS. Technical report, version 2.2, 2009.

[6] Thomas Morton and Jeremy LaCivita. Wordfreak: An open tool for linguistic annotation. In *Proceedings of NAACL-Demonstrations '03*, pages 17–18, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073427.1073436. URL http://dx.doi.org/10.3115/1073427.1073436.

[7] Petr Pajas and Jan Štěpánek. Recent advances in a feature-rich framework for treebank annotation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 673–680, Manchester, UK, August 2008. Coling 2008 Organizing Committee. URL `http://www.aclweb.org/anthology/C08-1085`.

[8] P. Paroubek and M. Rajman. Multitag, une ressource linguistique produit du paradigme d'évaluation. In *Actes de Traitement Automatique des Langues Naturelles*, pages 297–306, Lausanne, Suisse, 16-18 octobre 2000.

[9] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

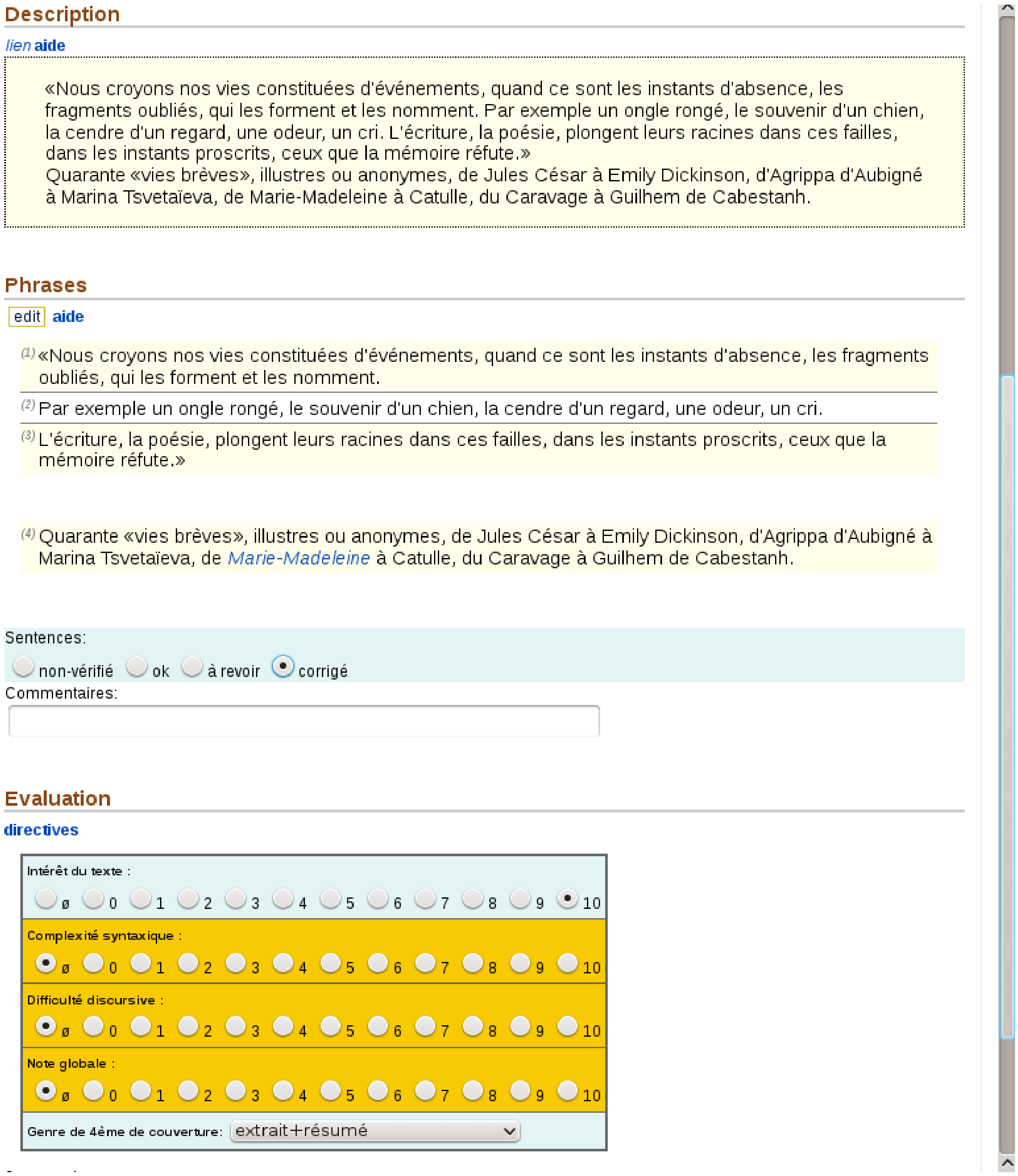
[10] S. Rauzy and P. Blache. Un point sur les outils du lpl pour l'analyse syntaxique du français. In *Actes du workshop ATALA 'Quels analyseurs syntaxiques pour le français ?'*, pages 1–6, Paris, France, 2009.

[11] Stéphane Rauzy and Philippe Blache. Robustness and processing difficulty models. a pilot study for eye-tracking data on the french treebank. In *Proceedings of Workshop on Eye-tracking and Natural Language Processing at The 24th International Conference on Computational Linguistics (COLING)*, 2012.

[12] Natalie Schluter and Josef van Genabith. Preparing, restructuring, and augmenting a french treebank: Lexicalised parsers or coherent treebanks? In *Proceedings of PACLING 07*, pages 200–209, 2007.

[13] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France, April 2012. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/E12-2021`.

[14] Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. Webanno: A flexible,web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*, pages 1–6, Stroudsburg, PA, USA, August 2013. Association for Computational Linguistics.



Description

lien **aide**

«Nous croyons nos vies constituées d'événements, quand ce sont les instants d'absence, les fragments oubliés, qui les forment et les nomment. Par exemple un ongle rongé, le souvenir d'un chien, la cendre d'un regard, une odeur, un cri. L'écriture, la poésie, plongent leurs racines dans ces failles, dans les instants proscrits, ceux que la mémoire réfute.»
Quarante «vies brèves», illustres ou anonymes, de Jules César à Emily Dickinson, d'Agrippa d'Aubigné à Marina Tsvetaïeva, de Marie-Madeleine à Catulle, du Caravage à Guilhem de Cabestanh.

Phrases

edit **aide**

(1) «Nous croyons nos vies constituées d'événements, quand ce sont les instants d'absence, les fragments oubliés, qui les forment et les nomment.

(2) Par exemple un ongle rongé, le souvenir d'un chien, la cendre d'un regard, une odeur, un cri.

(3) L'écriture, la poésie, plongent leurs racines dans ces failles, dans les instants proscrits, ceux que la mémoire réfute.»

(4) Quarante «vies brèves», illustres ou anonymes, de Jules César à Emily Dickinson, d'Agrippa d'Aubigné à Marina Tsvetaïeva, de *Marie-Madeleine* à Catulle, du Caravage à Guilhem de Cabestanh.

Sentences:
○ non-vérifié ○ ok ○ à revoir ● corrigé

Commentaires:

Evaluation

directives

Intérêt du texte :
○ ø ○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ● 10

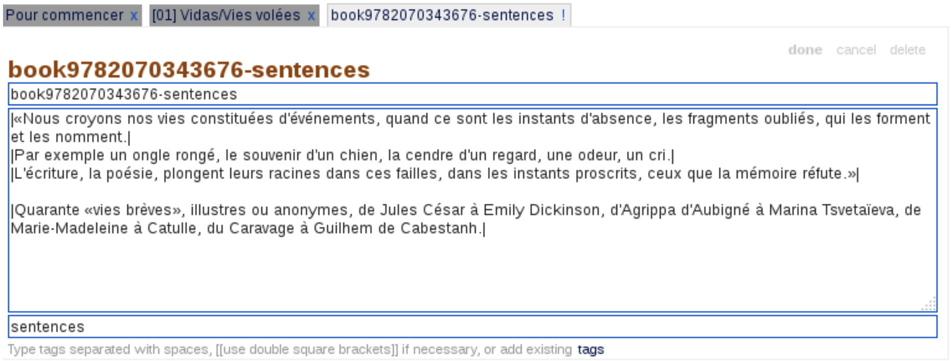
Complexité syntaxique :
● ø ○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10

Difficulté discursive :
● ø ○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10

Note globale :
● ø ○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10

Genre de 4ème de couverture: extrait+résumé

Figure 3: Text selection
(description of *Vidas/Vies volées*, Christian Garcin, edited by *Gallimard*)



book9782070343676-sentences

book9782070343676-sentences

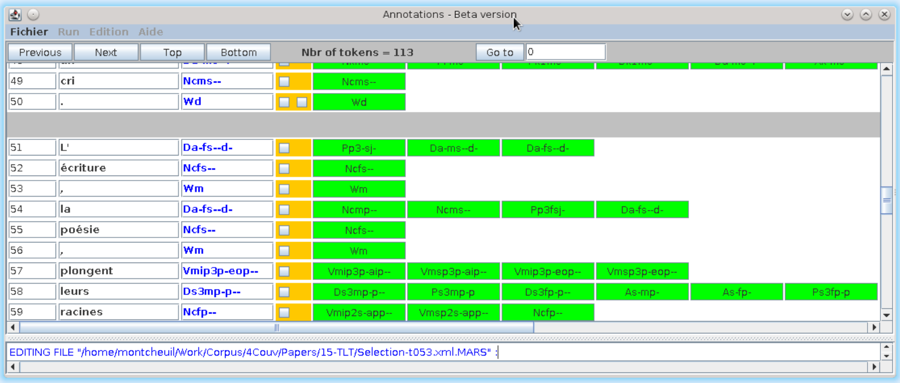
|«Nous croyons nos vies constituées d'événements, quand ce sont les instants d'absence, les fragments oubliés, qui les forment et les nomment.|
|Par exemple un ongle rongé, le souvenir d'un chien, la cendre d'un regard, une odeur, un cri.|
|L'écriture, la poésie, plongent leurs racines dans ces failles, dans les instants proscrits, ceux que la mémoire réfute.»|

|Quarante «vies brèves», illustres ou anonymes, de Jules César à Emily Dickinson, d'Agrippa d'Aubigné à Marina Tsvetaïeva, de Marie-Madeleine à Catulle, du Caravage à Guilhem de Cabestanh.|

sentences

Type tags separated with spaces, [[use double square brackets]] if necessary, or add existing tags

Figure 4: Editing sentences split and sections



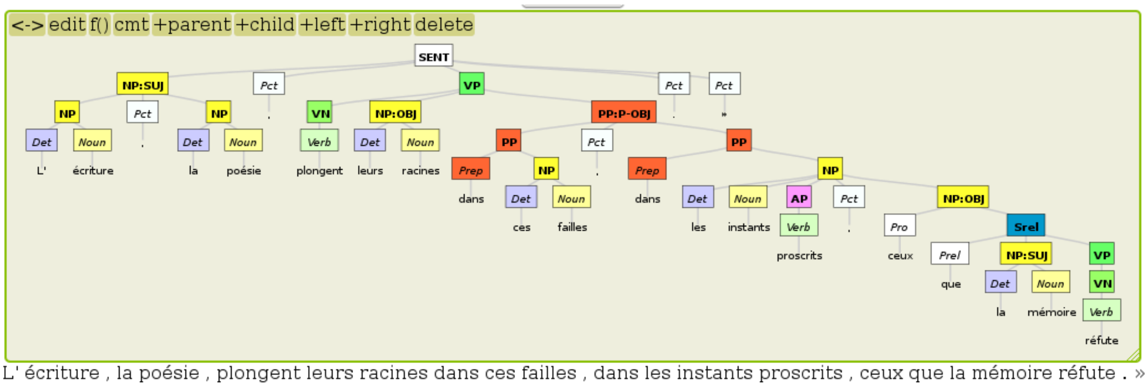
Nbr of tokens = 113

0

49	cri	Ncms--	□	Ncms--					
50	.	Wd	□ □	Wd					
51	L'	Da-fs--d-	□	Pp3-sj-	Da-ms--d-	Da-fs--d-			
52	écriture	Ncfs--	□	Ncfs--					
53	,	Wm	□	Wm					
54	la	Da-fs--d-	□	Ncmp--	Ncms--	Pp3fsj-	Da-fs--d-		
55	poésie	Ncfs--	□	Ncfs--					
56	,	Wm	□	Wm					
57	plongent	Vmip3p-eop--	□	Vmip3p-aip--	Vmsp3p-aip--	Vmip3p-eop--	Vmsp3p-eop--		
58	leurs	Ds3mp-p--	□	Ds3mp-p--	Ps3mp-p	Ds3fp-p--	As-mp-	As-fp-	Ps3fp-p
59	racines	Ncfp--	□	Vmip2s-app--	Vmsp2s-app--	Ncfp--			

EDITING FILE "/home/montcheuil/Work/Corpus/4Couv/Papers/15-TLT/Selection-t053.xml.MARS" :

Figure 5: Morphosyntactic tags correction



L' écriture , la poésie , plongent leurs racines dans ces failles , dans les instants proscrits , ceux que la mémoire réfute . »

Figure 6: Syntactic tree editor

Studying Consistency in UD Treebanks with INESS-Search

Koenraad De Smedt,[1] Victoria Rosén[1] and Paul Meurer[2]

[1]University of Bergen, [2]Uni Research Computing
E-mail: {desmedt|victoria|paul.meurer}@uib.no

Abstract

We demonstrate how treebank search may be helpful in examining the consistency with which annotations are applied, both within and across treebanks. Universal Dependency (UD) treebanks are used as examples.

1 Background

If annotation guidelines are to be universally applied to several treebanks, good search tools are necessary to retrieve and compare annotations both within and across treebanks. In a small scale study, we have performed a number of searches in the second release (version 1.1) of the Universal Dependencies (UD) treebanks.[1] These treebanks result from a large coordinated effort to create similarly annotated treebanks across several languages,[2] building on an initiative by the Universal Dependency Treebank (UDT) project [3], the development of Universal Stanford Dependencies [2] and other work.[3]

In order to facilitate the study of the UD treebanks by the research community, we have imported and indexed them for search in the INESS treebanking infrastructure [5]. Since these treebanks for 18 languages have a common annotation format, it is possible to search in all of them simultaneously with INESS-Search [4], an online tool with a powerful query language.[4] This tool makes it relatively simple to get an impression of the degree to which certain constructions are annotated in a parallel way across different languages in the UD treebanks, and the degree to which this annotation is consistently applied within each treebank. Our aim has

[1] Obtained from the LINDAT/CLARIN repository, http://hdl.handle.net/11234/LRT-1478, on June 30, 2015.

[2] The UD treebanks v1.1 cover 18 languages: Basque, Bulgarian, Croatian, Czech, Danish, English, Finnish, French, German, Modern Greek, Hebrew, Hungarian, Indonesian, Irish, Italian, Persian, Spanish and Swedish.

[3] See http://universaldependencies.github.io/docs/ for more history, bibliography and other information. All cited pages on this website were consulted on June 30, 2015.

[4] http://clarino.uib.no/iness

not been to perform any systematic and exhaustive evaluation, but to explore and demonstrate search techniques in an illustrative way.

2 Annotation guidelines for the UD treebanks

In the present study we wanted to search for annotations of multiword expressions (MWEs). The annotation guidelines on the UD documentation website do not provide a separate treatment of MWEs as such, but do say that there are three dependency relations that may be used for compounding: *compound*, *name* and *mwe*.[5]

It is not clear whether all *compound* constructions in UD are to be considered MWEs, but at least one subtype is commonly considered to be an MWE: verb-particle constructions [6]. These are to be annotated with the *compound:prt* dependency relation (where *prt* stands for *particle*).[6] For example, in English *shut down*, the *compound:prt* relation holds between the verb and its particle.

The *name* relation is to be used for "proper nouns constituted of multiple nominal elements",[7] for example *Hillary Rodham Clinton*. The structure is flat, with all words modifying the first one using the *name* label. This simple guideline is, however, amended by additional ones which are somewhat more difficult to interpret. The *name* annotation is only to be used when there is no clear syntactic modification structure. If there is one, regular syntactic relations are used. The given example *Río de la Plata* may be said to have an internal syntactic structure in Spanish, but this is not obvious when this name is used in English.

The *mwe* dependency relation is to be used for roughly the category of *fixed expressions* [6], with the exception of relations that should be annotated with the *compound* or *name* labels. The annotation is a "flat, head-initial structure, in which all words in the expression modify the first one using the mwe label".[8] An example is *as well as*, where *as* is the head and the other words are dependent on it through *mwe* relations.

3 Fixed expressions

An interactive search for the *mwe* relation in all treebanks was performed by selecting all UD treebanks and entering the query shown in (1) on the *Query* page of the online INESS-Search service. In this query, `>mwe` stands for the dependency relation from the variable `#x_` to the variable `#y_`. The metadata parameter `lang` matches the language of the treebank and allows us to map the distribution of search results for the different languages.

[5] `http://universaldependencies.github.io/docs/u/dep/compound.html`. This treatment of compounding is consistent with the Stanford Universal Dependencies proposal [2].

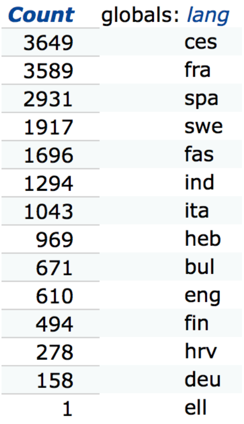
[6] `http://universaldependencies.github.io/docs/en/dep/compound-prt.html`

[7] `http://universaldependencies.github.io/docs/u/dep/name.html`

[8] `http://universaldependencies.github.io/docs/u/dep/mwe.html`

(1) `#x_ >mwe #y_ :: lang`

The results of this query are displayed in a table with a column for frequencies and one column per variable. When a variable name contains an underscore, its values are not displayed in the table. Thus, Table 1 displays only the distribution for the *mwe* relation per language (displayed with its ISO code).



Count	globals: *lang*
3649	ces
3589	fra
2931	spa
1917	swe
1696	fas
1294	ind
1043	ita
969	heb
671	bul
610	eng
494	fin
278	hrv
158	deu
1	ell

Table 1: Screenshot of search results for *mwe* obtained by query (1)

The total number of matches is 19300 in 14 languages. Languages in which the *mwe* relation does not occur (i.e. Basque, Danish, Hungarian and Irish), do not appear in the table. Clicking on a line in the table brings up a list of all matching sentences. Clicking on a particular sentence displays the analysis of that sentence.

The numbers for the various languages are absolute frequencies and thus not directly comparable, but it is remarkable that there is only one occurrence of an *mwe* relation in Greek. Clicking on the line for Greek revealed an *mwe* relation for ακόμα και "even", as shown on the lefthand side of Figure 1. A further search for this word pair with the query in (2), where the dot is the immediate linear precedence operator, revealed eight occurrences in the Greek UD treebank. The seven occurrences without *mwe* use advmod relations, as illustrated on the righthand side of Figure 1.

(2) `"ακόμα" . "και"`

Because the guidelines clearly specify that *mwe* should be used in a head-initial way, we decided to check this. To simplify matters, we checked this in *mwe* dependency relations with only one *mwe* dependent, and checked the linear precedence of the words. A search for such binary *mwe* relations with the query in (3), which can be paraphrased as "*mwe* relations with a head *x* and a dependent *y* where there is no *mwe* relation from the head to any *z* that is not identical with *y*." This resulted in 14478 matches in all UD treebanks, with the distribution shown in Table 2.

(3) `#x_ >mwe #y_ & !(#x_ >mwe #z & #z != #y_) :: lang`

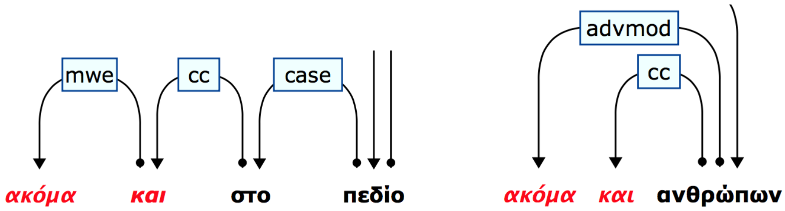
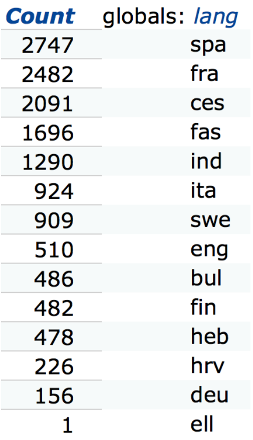


Figure 1: Screenshots of two of eight search results for ακόμα και



Count	globals: *lang*
2747	spa
2482	fra
2091	ces
1696	fas
1290	ind
924	ita
909	swe
510	eng
486	bul
482	fin
478	heb
226	hrv
156	deu
1	ell

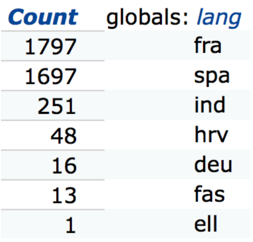
Table 2: Search results for (5)

Head-initial annotations of adjacent word pairs were retrieved by the query in (4) in which the head immediately precedes the dependent. This resulted in 9454 matches for adjacent word pairs which have the correct dependency direction.

Head-final annotations of adjacent word pairs were retrieved by the query in (5), resulting in 3823 matches which have the opposite, i.e. incorrect, dependency direction according to the guidelines. Only seven languages have such head-final *mwe* annotations. Their distribution is shown in Table 3. When these numbers are seen in relation to those in Table 2, we note that well over half of the French and Spanish binary *mwe* dependencies are head-final, compared to less than one percent of those in the Persian treebank. This merits further attention on the part of the treebank developers.

(4)
```
#x_ >mwe #y_ & #x_ . #y_
 & !(#x_ >mwe #z & #z != #y_) :: lang
```

(5)
```
#x_ >mwe #y_ & #y_ . #x_
 & !(#x_ >mwe #z & #z != #y_) :: lang
```

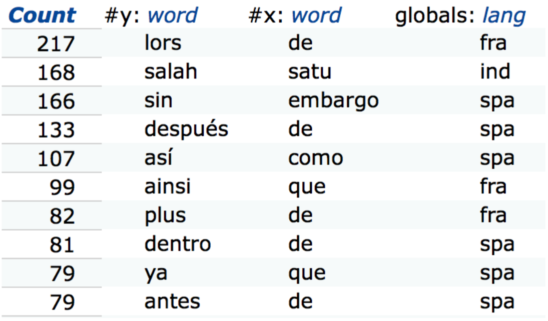


Count	globals: lang
1797	fra
1697	spa
251	ind
48	hrv
16	deu
13	fas
1	ell

Table 3: Search results for (5)

Table 4 shows the most frequent word pairs resulting from the query in (6), which is the same as that in (5) except that the variables `#x` and `#y` have no underscores, and therefore their values are included in the table. For some types, the dependency direction is consistently non-compliant, for instance for Spanish *sin embargo* "nevertheless". For others it is more evenly divided, as for Spanish *ya que* "since", with *ya* dominating *que* 78 times, whereas the opposite was found 79 times.

(6)
```
#x >mwe #y & #y . #x
 & !(#x >mwe #z & #z != #y) :: lang
```

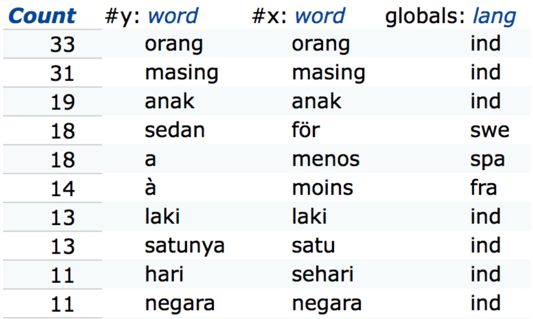


Count	#y: word	#x: word	globals: lang
217	lors	de	fra
168	salah	satu	ind
166	sin	embargo	spa
133	después	de	spa
107	así	como	spa
99	ainsi	que	fra
82	plus	de	fra
81	dentro	de	spa
79	ya	que	spa
79	antes	de	spa

Table 4: Top frequent types for (6)

(7)
```
#x >mwe #y & !(#y . #x) & !(#x . #y)
 & !(#x >mwe #z & #z != #y) :: lang
```

Furthermore, since fixed expressions are not supposed to have intervening material [6], the query in (7) was used to identify non-adjacent word pairs annotated with an *mwe* dependency relation. This yielded 1201 matches, the most frequent of which are displayed in Table 5. Some of these seem to be due to contractions,



Count	#y: word	#x: word	globals: lang
33	orang	orang	ind
31	masing	masing	ind
19	anak	anak	ind
18	sedan	för	swe
18	a	menos	spa
14	à	moins	fra
13	laki	laki	ind
13	satunya	satu	ind
11	hari	sehari	ind
11	negara	negara	ind

Table 5: Search results for (7)

as in French *au moins* “at least”, where *au* is a contraction of *à* and *le*, while others can be attributed to hyphenated reduplications, as in Indonesian *orang-orang* “people”. It is doubtful if Swedish *för . . . sedan* “. . . ago” can be considered a fixed expression, since another word or several words *must* intervene between *för* and *sedan*, but it could be a semi-fixed expression [1]. A *mwe* dependency between non-adjacent words may also indicate possible errors, for instance, German *wie auch auf* “as well as on” having a *mwe* dependency between the first and third words, whereas the first and second words are more likely candidates for a fixed expression.

4 Multiword names

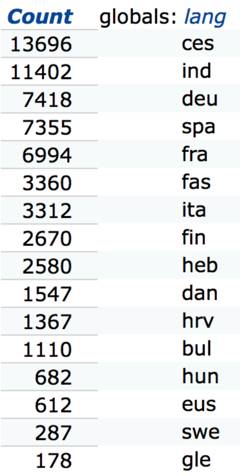
Proper nouns consisting of multiple nominal elements were searched for with the dependency *name* as shown in (8), producing the overview in Table 6. There were 64570 matches.

(8) `#x_ >name #y_ :: lang`

A name that is found frequently in many treebanks is *New York*, for which the dependency direction between *New* and *York* was mapped per language with the query in (9).

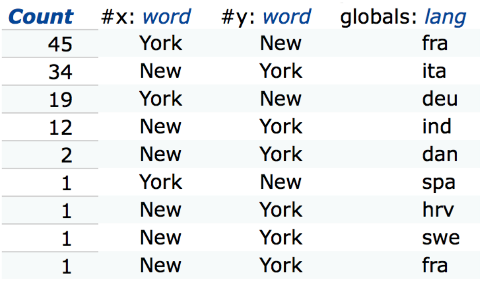
(9) `#x:[word="New|York"] >name #y:[word="New|York"] :: lang`

This resulted in 51 occurrences where *New* dominates *York* in Croatian, Danish, French, Indonesian, Italian and Swedish, in agreement with the guidelines, while 65 occurrences with the opposite dependency direction were found in French, German and Spanish, as shown in Table 7.



Count	globals: lang
13696	ces
11402	ind
7418	deu
7355	spa
6994	fra
3360	fas
3312	ita
2670	fin
2580	heb
1547	dan
1367	hrv
1110	bul
682	hun
612	eus
287	swe
178	gle

Table 6: Search results for *name* (8)



Count	#x: word	#y: word	globals: lang
45	York	New	fra
34	New	York	ita
19	York	New	deu
12	New	York	ind
2	New	York	dan
1	York	New	spa
1	New	York	hrv
1	New	York	swe
1	New	York	fra

Table 7: Search results for *New York* (9)

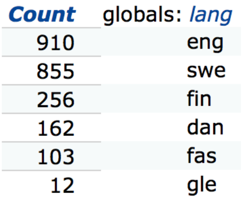
The *name* relation does not occur in the English and Greek UD treebanks. The English treebank uses the *compound* relation for names, with the last element as the head. Thus, there is a *compound* dependency from *York* to *New*. In the Greek treebank, there is an *amod* dependency to Νέα "New" from Υόρκη "York" and there is an *nmod* dependency to Χίλαρι "Hillary" from Κλίντον "Clinton", contrary to the example in the guidelines. In addition to this variation in dependency relations, the *mwe* relation is sometimes used in the annotation of names. In the Swedish treebank, the magazine name *Unesco Courier* uses *name*, but the name of a booklet, *Undervisning eller undergång* "Teaching or Doom", uses *mwe*. The Italian treebank annotates some names, for instance, *Scènes de la Vie privée*, with a mixture of *mwe* and *name* relations. Using *mwe* in multiword names does not seem consistent with the guidelines and creates difficulties in retrieving multiword names as distinct from fixed expressions.

Modifiers following names, such as titles and appositions,[9] are sometimes annotated as part of the name, sometimes not. In the Danish treebank, the last word in the string *Stefan Fryland, formand* "Stefan Fryland, leader" is annotated with a *name* dependency, whereas a similar construction in Spanish is annotated with the *appos* dependency, for instance, *Jerónimo Martín Caro y Cejudo [. . .], humanista* ". . . , humanist". German uses *appos* dependencies for modifiers preceding names, for instance for *Inhaber Michael Walther* "Proprietor . . . ", whereas Swedish uses the *det* relation for *professor* or for *författaren* "the author" preceding a name. The use of these relations deserves further investigation as it may involve additional distinctions.

5 Verb-particle constructions

Verb-particle constructions were searched for by means of the query in (10), which yielded 2298 matches in five languages. Their distribution is shown in Table 8.

(10) `#x_ >compound:prt #y_ :: lang`



Count	globals: lang
910	eng
855	swe
256	fin
162	dan
103	fas
12	gle

Table 8: Search results for *compound:prt* (10)

In the German treebank, the dependency relation *mark* is used for such constructions, for instance, *wuchs . . . auf* "grew up", *teilte . . . mit* "informed". This is not in accordance with the way the relation *mark* is described in the guidelines: "the word introducing a finite clause subordinate to another clause".[10] The Hungarian treebank seems to use *compound:preverb* for the verb-particle construction.

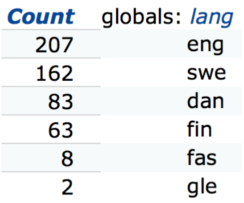
Verb-particle constructions can be discontinuous, for instance, *blow something up*. Such cases were searched for in all treebanks with the search expression in (11) resulting in the overview in Table 9. The results indicate that all treebanks which have the *compound:prt* relation have discontinuous constructions.

(11) `#x_ >compound:prt #y_ & !(#x_ . #y_) & !(#y_ . #x_) :: lang`

In the Danish treebank, the dependency relation *compound:prt* is not only used for phrasal verbs, but also between the elements of (discontinuous) circumpositions such as the frequent *for . . . siden* ". . . ago", as in *for to år siden* "two years ago".

[9] http://universaldependencies.github.io/docs/u/dep/appos.html

[10] http://universaldependencies.github.io/docs/u/dep/mark.html



Count	globals: *lang*
207	eng
162	swe
83	dan
63	fin
8	fas
2	gle

Table 9: Search results for discontinuous *compound:prt* (11)

6 Conclusion

We have reported on a small scale study which intends to show the usefulness of INESS-Search for getting an impression of the consistency of annotation both within and across treebanks. INESS-Search is available as an online tool in the INESS treebanking infrastructure. We selected the UD treebanks because they have common annotation guidelines. Our objective has not been to identify all possible phenomena related to MWEs and compounding in the UD treebanks, nor to provide alternative recommendations for the UD treatment of such phenomena.

We have made some preliminary assessments for the three specific dependency relations *mwe, name* and *compound:prt*. Systematic searches for these three relations with INESS-Search produced tables of distributions of annotations over several treebanks. We also performed a few sample searches for specific strings and constructions. Our findings indicate that the use of these relations in various treebanks does not always adhere to the guidelines. We also found seemingly unmotivated discrepancies even within treebanks.

For the *mwe* and *name* relations, there is sometimes a lack of consistency with respect to which part is the head. The guideline about head-initial annotation should be easy to follow, but is not followed in almost 3 out of 10 cases for binary *mwe* relations. For multiword names, there are discrepancies as to whether to annotate them with *name* or other syntactic relations, even for the same name in different treebanks. Depending on the labels used, it may therefore be more or less difficult to retrieve multiword names from the treebanks. Appositions seem to be annotated with at least three different dependencies even though they are very similar constructions across languages.

We could not systematically investigate where certain dependencies are *not* used when they should have been. In order to get an impression of the latter situation, we have merely performed a few sample searches with labels other than *mwe, name* and *compound:prt* without attempting to be exhaustive.

We hope that the present study will serve as a pilot for broader and more systematic investigations of the UD treebanks, and will thus benefit the quality of their annotation in future versions. Finally, for many other treebanks it is possible to use INESS-Search, provided the treebanks are uploaded to INESS and indexed.

Acknowledgments

This work has been supported by the INESS project funded by the Research Council of Norway and the University of Bergen, and by the PARSEME COST Action (IC1207) funded by the EC.

References

[1] Timothy Baldwin and Su Nam Kim. Multiword expressions. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing*, chapter 12. CRC Press, Boca Raton, FL, USA, 2nd edition, 2010.

[2] Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal Stanford Dependencies: a cross-linguistic typology. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA).

[3] 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 (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[4] Paul Meurer. INESS-Search: A search system for LFG (and other) treebanks. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG '12 Conference*, LFG Online Proceedings, pages 404–421, Stanford, CA, 2012. CSLI Publications.

[5] Victoria Rosén, Koenraad De Smedt, Paul Meurer, and Helge Dyvik. An open infrastructure for advanced treebanking. In Jan Hajič, Koenraad De Smedt, Marko Tadić, and António Branco, editors, *META-RESEARCH Workshop on Advanced Treebanking at LREC2012*, pages 22–29, Istanbul, Turkey, 2012.

[6] Ivan Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *Lecture Notes in Computer Science. Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, volume 2276, pages 189–206. Springer, 2002.

Building a Dependency Parsing Model for Russian with MaltParser and MyStem Tagset

Kira Droganova

Faculty of Humanities
Higher School of Economics
E-mail: kira.droganova@gmail.com

Abstract

The paper describes a series of experiments on building a dependency parsing model using MaltParser, the SynTagRus treebank of Russian, and the morphological tagger Mystem. The experiments have two purposes. The first one is to train a model with a reasonable balance of quality and parsing time. The second one is to produce user-friendly software which would be practical for obtaining quick results without any technical knowledge (programming languages, linguistic tools, etc.).

1 Introduction

There was a number of experiments on building dependency parsing models for Russian using MaltParser conducted previously. MaltParser was suggested and described by Nivre (Nivre et al. [1]). They did not include Russian in the languages used for experiments when describing general methodology and evaluation, however the subsequent experiments were performed on the SynTagRus treebank of Russian (Nivre et al. [2]) which currently contains 860 000 words. During the training of the model both lexical and morphological features were used. Further work presented by Sharoff (Sharoff, Nivre [3]) describes pipeline and tools for processing Russian texts. This software is represented as a set of scripts which need to be put together before use. All previously reported experiments were carried out involving TnT for POS-tagging and MaltParser for syntactic parsing.

In our approach, we use morphological information as the only input. The morphological tagger Mystem (Segalovich [4]) was designed specifically for Russian and has extremely useful settings which allow to make disambiguation by context. Moreover, the original morphological tagset of SynTagRus, ETAP-3 (Iomdin et al. [5]), is closer to Mystem than to TnT tagger. Since this has a direct influence on the quality of the parsing, our experiments was conducted using Mystem.

2 Approach

The project was divided into three levels: POS-tagging, training data, and tuning MaltParser settings. The final models were trained by combining the best results for each level. Thus, the pipeline was:

1. To prepare Mystem annotation using SynTagRus.

Mystem annotation were obtained using two methods. The original tagset of SynTagRus was mapped to the Mystem tagset using a conversion table in order to improve the accuracy of the tagging. There are certain mismatches in Mystem and ETAP3 tagsets, for example, personal pronouns are tagged as nouns in SynTagRus, there is no predicatives, parentheses and some other POS-tags. Moreover, SynTagRus includes multitokens (multi-word prepositions, adverbs, etc.). All these variations could affect parsing quality relating to actual data.

An alternative approach is to re-annotate SynTagRus, i.e. to get a certain word form from SynTagRus and send it to Mystem. The result is generally more accurate, but the composites (e.g. general-major 'Major General') often get recognized as two separate tokens by Mystem and as one token by ETAP 3, and vice versa, and this results in erroneous output. Therefore at present the quality of the models is much worse compared to the first approach.

In the future we are planning to apply Mystem (a version with disambiguation) directly both during training and during annotation of new data. We expect the reannotation to help to produce more accurate tags for composites during training and obtain better results.

2. To prepare training data by converting SynTagRus into conll-file.

SynTagRus was split into three parts: the training set (80%), the development test set (10%) and the final test set (10%). The original SynTagRus format (Iomdin et al. [5]) was converted into conll-file [6] using a convertion scheme.

Figure 1 provides an example of conversion scheme. Lines 1 and 3 provide information about SynTagRus structure, lines 2 and 4 relate to conll layer, which is the data format for MaltParser. The conversion scheme was developed for the purpose of transforming SynTagRus data into training data in conll format. For example, value " root" of the attribute "DOM" indicates the head of the sentence and should be converted into zero in the 6th conll layer position and into "root" in the 7th position. First three positions are typically converted inalterably. Concerning conll layer positions from 4 to 6, variations are allowed, such as for instance, part of speech and morphological data separation.

There was a number of experiments on a size of the data, punctuation marks and content of a field [6] performed previously. The most valuable experiments were performed on CPOSTAG (Coarse-grained part-of-speech tag) and POSTAG (Fine-grained part-of-speech tag) fields, where the 'three letter models' were trained. These models have three letters from the word ending in CPOSTAG or POSTAG.

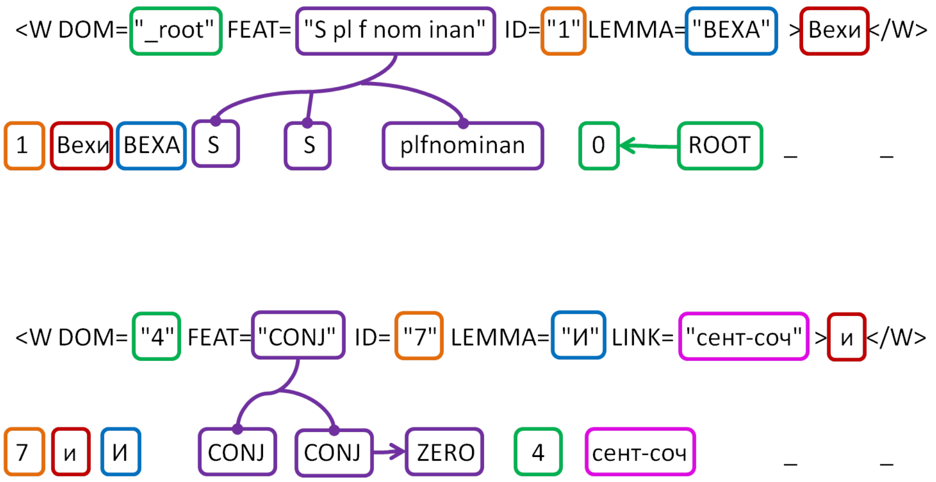

<W DOM= "_root" FEAT= "S pl f nom inan" ID= "1" LEMMA= "ВЕХА" > Вехи </W>
1 Вехи ВЕХА S S plfnominan 0 ROOT _ _
<W DOM= "4" FEAT= "CONJ" ID= "7" LEMMA= "И" LINK= "сент-соч" > и </W>
7 и И CONJ CONJ ZERO 4 сент-соч _ _

Figure 1: An example of conversion scheme

3. To train the models with different settings.

A large number of experiments was conducted using different types of projective and non-projective algorithms. Non-projective algorithms allow branches to cross as opposed to projective ones. The most valuable results have been achieved while using pseudo-projective transformations provided by MaltParser functionality.

3 Results

Results were measured with MaltEval using Labeled Attachment Score and Unlabeled Attachment Score evaluation metrics [7]. Accuracy reaches 80.3% by LAS and 87.5% by UAS for the best model with punctuation. Error evaluation is based on approach described by Toldova [8] adapted for the purpose of these experiments. The classification comprises 5 error types:

- Type 1 — wrong root predicted.
- Type 2 — wrong head predicted.
- Type 3 — wrong label predicted.
- Type 4 — wrong head predicted (acceptable error).
- Type 5 — wrong label predicted (acceptable error).

Type 1 is common for compound sentences longer than ten words. Normally, if the sentence has type 1 error, it has many type 2 errors. A large amount of type 3 errors is due to special aspects of syntactic relations in Russian. 65 types of syntactic relations are used in SyntagRus and this results in lack of examples for some rarely used relations. Due to the so-called 'free word order' in Russian, wrong labels appear almost every time the head is predicted incorrectly. There are however more reasons for false labeling.

Types 4 and 5 are not indicative as they do not have significant effect on parsing result. These errors appears when individual type of syntactic relation is predicted as a general type or when predicted relation is an alternative version of tagging.

Future work includes a deeper analysis of the training data (word frequency, uncommon words), experiments with transforming syntactic relations into more simple structure and with special attention to the universal dependencies.

4 Conclusion

The paper presents the first results of building the dependency parsing model as the first step to produce a digital resource for linguists. Using all of the original SynTagRus syntactic relations and Mystem POS-tagging the model accuracy reaches up to 80.3% by LAS and 87.5% by UAS. Even though the results reported by Sharoff and Nivre [3] are slightly better (for SynTagRus tags: LAS 83.4, UAS 89.4), they are not comparable to ours due to the differences in training data and impossibility to replicate the experiments on the same dataset.

5 Acknowledgements

The author is grateful for computational capabilities provided by Andrey Kutuzov and mail.ru group.

References

[1] Nivre, Joachim, Hall, Jonah, Nilsson, Jens, Chanev, Atanas, Eryigit, Gülsen, Kübler, Sandra, Marinov, Svetoslav, Marsi, Erwin (2007) MaltParser: A language independent system for data-driven dependency parsing. Natural Language Engineering, 13, pp. 95-135.

[2] Nivre, Joachim, Boguslavsky, Igor M., Iomdin, Leonid L (2008) Parsing the SYNTAGRUS Treebank of Russian, In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 641–648.

[3] Sharoff, Serge, Nivre, Joachim (2011) The proper place of men and machines in language technology. Processing russian without any linguistic knowledge', In Computational Linguistics and Intelligent Technologies: Proceedings of the International Conference Workshop Dialogue 2011). Vol. 10 (17), 2011. Moscow: RGGU, pp. 657-670.

[4] Segalovich, Ilya (2003) A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In MLMTA-2003. (URL: https://tech.yandex.ru/mystem/)

[5] Iomdin, Leonid, Petrochenkov, Vyacheslav, Sizov, Viktor, Tsinman Leonid (2012) ETAP parser: state of the art. In Computational linguistics and intellectual technologies. Proceedings of International Workshop Dialogue 2012. Vol. 11 (18), 2012. Moscow: RGGU, pp. 830-848.

[6] Depparse Wiki (URL:http://depparse.uvt.nl/DataFormat.html)

[7] Nilsson, Jens, User Guide for MaltEval 1.0 (beta), 2014 (URL:http://www.maltparser.org/malteval.html)

[8] Toldova, Svetlana, Sokolova, Elena, Astafiyeva, Irina, Gareyshina, Anastasia, Koroleva, Anna, Privoznov, Dmitry, Sidorova, Evgenia, Tupikina, Ludmila, Lyashevskaya, Olga (2012) Ocenka metodov avtomaticheskogo analuza teksta 2011-2012: Sintaksicheskie parsery russkogo jazyka [NLP evaluation 2011-2012: Russian syntactic parsers]. In Computational linguistics and intellectual technologies. Proceedings of International Workshop Dialogue 2012. Vol. 11 (18), 2012. Moscow: RGGU, pp. 797-809.

Identifying Compounds: On The Role of Syntax

Murhaf Fares, Stephan Oepen and Erik Velldal

Language Technology Group, Department of Informatics
University of Oslo
Email: {murhaff|oe|erikve}@ifi.uio.no

Abstract

In this work, we return to a foundational problem related to the interpretation of nominal compounds (in English) that has received comparatively little attention in past research, viz. the identification of instances of the compound construction. We review techniques proposed for this task previously and contrast different approaches along three dimensions of variation, including the contrast of assuming part of speech annotations only vs. using full constituent structure. A first set of quantitative and qualitative experimental results suggest that syntax-based compound identification leads to far better results, at least where gold-standard constituent structures are available.

1 Introduction

In an email among the authors of this paper, one said: "I got a *kitchen update* from Joe." *Kitchen update*, albeit uncommon, is a valid example of nominal compounding in English, where a more typical example could be, say, *apple juice* or *lung cancer*. Downing [4] refers to nominal compounds as "noun plus noun compounds" and adopts the definition by Li [13] as "the concatenation of any two or more nouns functioning as a third nominal." Similarly, our approach defines noun compounds as constructions consisting of two or more nouns that stand in a head–modifier relation.

One of the characteristics of noun compounds is their semantic unpredictability. The aforementioned compound *kitchen update*, for example, may refer to an update about the kitchen status or an update (about whatever) that happened to be given in the kitchen. Furthermore, compounding is a very frequent and productive linguistic process: Baldwin and Tanaka [1] report that 2.6% of the words in the written portion of the British National Corpus (BNC; Burnard [2]) and 3.9% of the Reuters corpus (Rose et al. [20]) are contained in noun–noun compounds. This indicates that a principled and systematic treatment of these constructions will be of potential importance to a wide range of Natural Language Processing (NLP) tasks.

Lauer and Dras [11] identify three tasks related to noun compounds: (1) detection or identification of noun compounds, (2) syntactic analysis of the internal structure, i.e. left vs. right bracketing of compounds with more than two constituents, and finally (3) interpretation of the semantic relation holding between the constituents of the compound. The task of noun compound interpretation has been the focus of many studies (Tratz and Hovy [21], Nakov [16], Ó Séaghdha and Copestake [18]), including several SemEval shared tasks (Girju et al. [7], Butnariu et al. [3], Hendrickx et al. [9]). The bracketing task has also received some attention, either as a separate task (Nakov [16], Pitler et al. [19]) or as part of parsing noun phrases (Vadas and Curran [23]). However, the task of noun compound identification has not received as much attention. This paper presents careful analysis and experimentation directed at the identification task, demonstrating the benefit of using syntactic information. We believe that more accurate noun compound identification will have an effect on the other two tasks of bracketing and interpretation. Further, the three tasks become even more interdependent in the context of our efforts to automatically construct a data set of noun compounds with their semantic interpretation (we will elaborate more on the context of this research in § 6).

In § 2, we briefly review previous work on noun compound identification. In § 3 we define three main variables for noun compound identification strategies. In § 4 we present our approach and experimental setup. In § 5 we report the results of our experiments with a brief analysis. We reflect on the results analysis in § 6, and, finally, in § 7 we conclude the paper.

2 Background

Variations of the heuristic suggested by Lauer [12] comprise some of the most widely used symbolic approaches to noun compound identification (Girju et al. [6], Ó Séaghdha [17], Tratz and Hovy [21]). Lauer [12] defines noun compounds as consecutive pairs of so-called "sure nouns"—nouns that are unambiguous with respect to their part-of-speech (PoS) tags—that are not preceded and not followed by other nouns. Several studies rely on variations of the heuristic of Lauer without mention of the restriction to unambiguous nouns (e.g. Tratz and Hovy [21]). Lauer [12] reports a high precision of 97.9% on a set of 1,068 candidate noun compounds from the Grolier Multimedia Encyclopedia, where an important factor presumably is his limitation of candidate compound constituents to unambiguous nouns.

Lapata and Lascarides [10] evaluated the heuristic of Lauer on the BNC by inspecting a sample of 800 noun sequences classified as valid compounds and report an accuracy of 71%, which is substantially lower than the original results by Lauer [12]. They mention PoS tagging errors when discussing these results.

In the same article, Lapata and Lascarides [10], also introduce statistical models (based on C4.5 decision tree and naïve Bayes learners) to identify noun compounds. They train and test the models on 1,000 noun sequences that occur only once in the BNC, and experiment with different combinations of features and learners. Their best model attains an accuracy of 72.3%. In addition to surface form statistics, Lapata and Lascarides [10] use PoS tag information, making it similar to the heuristic of Lauer in terms of the type of information used.

Importantly, Lauer [12] already points out that "there is no guarantee that two consecutive nouns form a compound." For example, bare direct and indirect nominal objects of a transitive verb can occur consecutively without forming a noun compound. In fact, some of the studies that used the heuristic of Lauer resorted to manual inspection of the extracted candidate noun compounds to exclude false positives (Girju et al. [7], Ó Séaghdha [17]). In the present paper we investigate the use of syntactic information to identify noun compounds. As explained in § 3, we expect that a richer linguistic representation may enable one to exclude some of the false positives and include some of the missing false negatives.

3 Noun Compound Identification Strategies

In order to state the problem and our approach more precisely, we define three dimensions of noun compound identification strategies. One dimension is the type of linguistic information used to detect noun compounds, namely PoS tags (*PoS-based*) and syntax trees (*syntax-based*). A second dimension regards the treatment of proper nouns (NNPs), where we can define three options: (a) Simply treat proper nouns like common nouns (i.e. no special treatment), (b) exclude all noun sequences that contain proper nouns or (c) exclude noun sequences that are headed by a proper noun (assuming that the head is always the right-most word in the sequence). We refer to those three strategies as NNP^*, NNP^0 and NNP^h, respectively. A third dimension regards the number of constituents (i.e. nouns) within the noun compound. This is dependent on the type of linguistic information we use to identify noun compounds. In the PoS-based approach, we distinguish between *binary* and *n-ary* strategies for compound identification, where the former identifies noun+noun compounds and the latter identifies compounds that have $n >= 2$ constituents. In the syntax-based approach, we also distinguish between binary and n-ary compounds, but additionally taking into consideration that the bracketed structure of n-ary compounds is available. Hence, we can decompose n-ary noun compounds, where $n > 2$, into 'sub-compounds' including binary ones. We will explain the abovementioned dimensions using the following example sentence from the venerable Wall Street Journal (WSJ) section of the Penn Treebank (PTB; Marcus et al. [14]):

... Nasdaq$_{NNP}$ bank$_{NN}$ index$_{NN}$$_{,}$, which$_{WDT}$ tracks$_{VBZ}$ thrift$_{NN}$ issues$_{NNS}$ ...

First, under a PoS-based binary strategy we will extract *thrift issues*, while an n-ary strategy will extract both *thrift issues* and *Nasdaq bank index*. As for the proper noun treatment, an NNP^0 strategy would exclude *Nasdaq bank index* but NNP^h would not because the proper noun *Nasdaq* is not in the head position. In the syntax-based approach, the same rule for NNP treatment would apply, but there

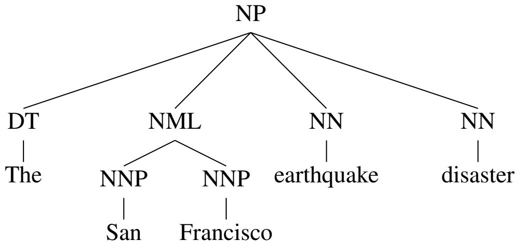


Figure 1: Internal noun phrase structure

will be more binary compounds, namely *bank index*, as syntax gives access to the internal structure of the compound *Nasdaq bank index*.

In our experiments we compare the PoS-based and syntax-based approaches for both binary and *n*-ary compounds, and NNP^0 and NNP^h for proper noun treatment.

4 Syntax-based Identification

The PoS-based strategy for noun compound identification requires a sequence of nouns that are not preceded and not followed by other nouns. With richer linguistic representations, such as syntactic trees, the definition of noun compounds goes one step further; the sequence of nouns is also a sequence of leaf nodes in the parse tree, hence the definition of a noun compound becomes a sequence of noun leaf nodes that are dominated by the same parent node—more specifically the same noun phrase parent node (we will amend this definition when we introduce the actual syntactic representation used in our experiments). The requirement of a single parent node stems from the fact that noun compounds act as one nominal, hence their constituents cannot belong to two different phrases.

In order to compare the PoS- and syntax-based strategies, we use the English part of the Prague Czech–English Dependency Treebank 2.0 (PCEDT; Hajič et al. [8]) which contains the WSJ section of the PTB. We chose to use the PCEDT because it includes the internal noun phrase annotations introduced by Vadas and Curran [22], whereas the 'original' PTB leaves the noun phrases flat.

Figure 1 shows an example of the internal annotation of noun phrases in the PCEDT. NML stands for *nominal modifier left-branching* and is one of the nodes introduced by Vadas and Curran [22]. The right-branching noun phrases were left unannotated. Our definition of noun compounds above requires leaf nodes to have an identical parent node, but in Figure 1 we see that *San Francisco* has a different parent node from the *earthquake disaster*, therefore in the implementation of syntax-based noun compound identification we make an exception for the identical-parent condition when the parent node is of type NML. In concrete terms, this means that we extract the following three compounds from the structure in Figure 1:

The ((San Francisco) (earthquake disaster)).

	PoS-Based				**Syntax-Based**			
	Binary		***N*-Ary**		**Binary**		***N*-Ary**	
	NNP^0	**NNP^h**	**NNP^0**	**NNP^h**	**NNP^0**	**NNP^h**	**NNP^0**	**NNP^h**
Tokens	27677	33167	30296	39429	29535	36441	34151	42835
Types	15128	18766	17167	23704	15853	20018	19469	25021

Table 1: Total number of noun compounds in PTB WSJ

Note that even though we make an exception for the identical-parent condition for NMLs, we still preserve their (left) bracketing constraints, hence, a compound like *Francisco earthquake* will not be extracted from the example phrase above.

5 Results and Discussion

In order to compare the PoS- and syntax-based approaches we experiment with detecting noun compounds in the full PTB WSJ in the PCEDT with eight different configurations as shown in Table 1, which provides total counts of compound instances (*tokens*) and the numbers of distinct strings (*types*).[1] In all configurations, the syntax-based strategy extracts more compounds than the PoS-based one, and that is because the former has access to the internal structure of the noun compounds and can therefore extract binary compounds out of n-ary ones where $n > 2$. Furthermore, in the binary setup, the PoS-based strategy is limited to strictly two consecutive nouns. The sequence $board_{NN}$ $meeting_{NN}$ $yesterday_{NN}$, for example, is not considered by the binary PoS-based strategy because it contains three consecutive nouns, whereas the syntax-based strategy extracts the sub-compound *board meeting*. Apart from this, the mere numbers do not tell us much in the absence of gold-standard data—to the best of our knowledge there is no gold-standard data set for noun compound identification. We therefore manually inspected a total of 100 random binary NNP^h compounds; 50 of which are only detected by the PoS-based strategy and 50 that are only detected by the syntax-based strategy.

Of the first set, 28 instances include a percent sign which is tagged as noun (NN) in the PTB, e.g. *% drop* in "… and a 4% drop in car loadings." In fact, *% stake* and *% increase* are among the top ten most frequent noun compounds identified by the PoS-based strategy, which is unsurprising given the WSJ domain. Such cases are easily excluded in the syntax-based strategy because the percent sign and the following noun belong to different constituents. We also identified five compounds that are due to annotation errors in the PTB on the PoS tag level, but not the syntax level. For example the tag NNS (plural noun) on the verb *amounts* in "one day's trading amounts to $7.6 billion". We also identified subtler annotation errors like annotating the adjective *in vitro* as preposition (IN) and noun (NN), which led the PoS-based strategy to extract *vitro cycles* as a compound in "… after only two in

[1] Note that no linguistic pre-processing (e.g. down-casing or stemming) was applied when calculating the type counts reported in Table 1.

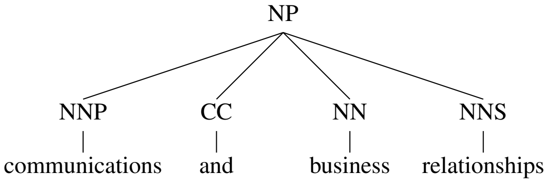


Figure 2: Coordination structure

vitro cycles". The remaining instances involve nouns that are not dominated by the same parent node. There are several linguistic constructions that may lead to such errors, such as the objects of a transitive verb and temporal modifiers like *today* and *yesterday* (tagged as nouns rather than adverbs in the PTB).[2] In sum the 50 compounds detected by only the PoS-based strategy are invalid noun compounds, which suggests that the syntax-based strategy succeeds in excluding some of the false positives referred to by Lauer [12].

Of the 50 noun compounds detected by the syntax-based strategy only, there are 38 compounds that were extracted from other compounds with more than two constituents—cases which could not have been identified by the binary PoS-based strategy. Furthermore, we see seven compounds that are either followed or preceded by other nouns. Such cases are also unidentifiable by the PoS-based strategy because it requires pairs of nouns not surrounded by other nouns. We also found four annotation errors where left-branching noun phrases were annotated as right-branching, for example in the phrase *San Diego home*, which leads to extraction of *Diego home* as a compound. The results analysis revealed that the syntax-based strategy includes arguably incorrect noun compounds when a noun is preceded by a coordinated phrase with noun conjuncts such as "communications and business relationships" in Figure2. The syntax-based strategy extracts *business relationships*, but this can be either incorrect or incomplete extraction given the nature of coordination structures as we will discuss in the following section.

The results analysis also revealed that our implementation of the identical-parent condition was not fine-grained enough to preserve the left bracketing information in some NML constituencies. For example, in Figure3 our implementation wrongly extracted the compound *development expenses*. In the following section we report the number of compounds extracted with a finer-grained implementation of the heuristic that handles such errors.

[2] According to the Part-of-Speech Tagging Guidelines of the PTB; "The temporal expressions yesterday, today and tomorrow should be tagged as nouns (NN) rather than as adverbs (RB). Note that you can (marginally) pluralize them and that they allow a possessive form, both of which true adverbs do not." See `http://groups.inf.ed.ac.uk/switchboard/POS-Treebank.pdf`

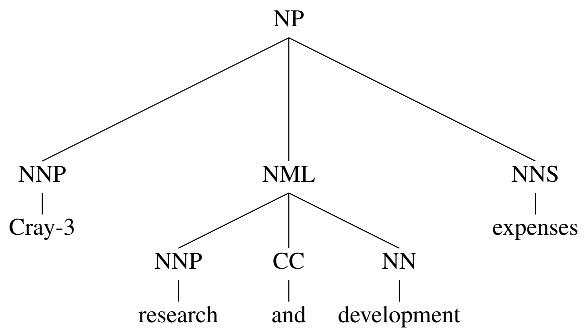


Figure 3: Coordination structure: Left-branching

6 Reflections

As shown in § 5, extracting noun compounds that are partially contained in nominal coordinate structures calls for careful treatment. In order to handle coordinate constructions properly, we need to distinguish between distributive and non-distributive (collective) coordinate structures. Consider the following coordinate constructions:

i Business and nursing programs

ii Research and development expenses

The first construction can be considered distributive and could be paraphrased as *business programs* and *nursing programs*. The second construction, however, is arguably non-distributive, which means that the two nominal conjuncts *research and development* 'jointly' modify the noun *expenses*—though it is also possible that the construction is referring to *research expenses* and *development expenses*, but we will assume that it is clearly non-distributive for the sake of argument. Given this distinction between distributive and non-distributive coordinate structures, it would in principle be possible to extract noun compounds from distributive coordinate structures, as we did with *business and nursing programs*. In practice, however, the PTB annotation does not distinguish between distributive and non-distributive coordinate structures, therefore we decided conservatively to exclude all noun compounds that are part of coordinate structures.

We further refined our implementation of the syntax-based identification heuristic to ensure that left-branching noun phrases are handled correctly. Consider the phrase "regional wastewater system improvement revenue bonds" in Figure4, which includes an adjectival modifier as part of the initial compound; according to our definition of noun–noun compounds (as strictly nominal sequences), the only compound that can be extracted from this phrase is *revenue bonds*. Given underspecified bracketing information within the first NML constituent, extracting *wastewater system* might be incorrect because, arguably, *wastewater* in this construction may be modified by *regional*, as shown in the following bracketing:

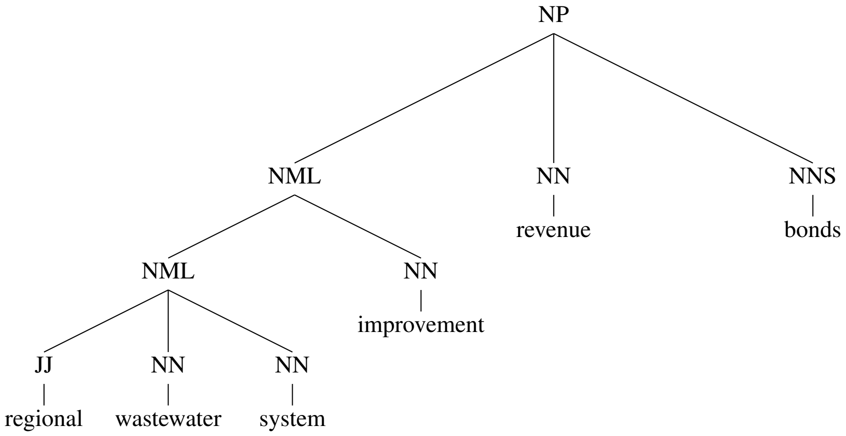


Figure 4: Complex left-branching noun phrase

((((regional wastewater) system) improvement) (revenue bonds))

Our refined implementation of the syntax-based heuristic, which also excludes all noun compounds that are part of a coordinate structure, identifies $33,095$ binary NNP^h compounds and $38,925$ n-ary NNP^h compounds, comparable in number to the PoS-based method (which would extract *some* compounds from both the conjoined modifier and adjectival modification structures of Figures 3 and 4). However, the trends regarding false positives and false negatives observed in the results analysis of § 5 apply with equal force to this more conservative parameterization of our syntax-based heuristics. We adopt this set of noun compounds as basis for our on-going work to automatically construct a data set of noun compounds with semantic relations based on the so-called PCEDT functors and noun senses and arguments in NomBank (Meyers et al. [15]).

7 Conclusion and Outlook

In this paper we presented two approaches to noun–noun compound identification, syntax-based and PoS-based. We identified three dimensions on which approaches to noun compound identification may vary. Our results and analysis suggest that achieving high-quality noun compound identification requires linguistic representations at least at the level of syntactic structure. We also show, however, that complex cases that include coordinate structures may require even richer linguistic annotations.

One of the challenges for quantifying the accuracy of the different identification strategies is the lack of gold-standard evaluation data. We therefore opted for manual inspection of the extracted compounds, which in turn led to gradual improvement in our implementation of the syntax-based identification heuristic.

In future work, we seek to extend our investigation into the utility of syntactic structure for the task of compound identification in two ways; by (a) evaluating the recent re-annotation of the WSJ Corpus in DeepBank (Flickinger et al. [5]) as a candidate gold standard, and by (b) gauging the effects on compound identification accuracy when moving from gold-standard syntactic structures to those available from state-of-the-art syntactic parsers. Also, we have started to combine our high-quality compound identification over PTB trees with thematic annotations over the same underlying text from resources like PCEDT and NomBank, aiming to fully automatically create comprehensive and high-quality gold-standard data for the thematic interpretation of relations among compound members.

References

[1] Timothy Baldwin and Takaaki Tanaka. Translation by machine of complex nominals. Getting it right. In *Second ACL Workshop on Multiword Expressions: Integrating Processing*, page 24–31, Barcelona, Spain, 2004.

[2] Lou Burnard. Reference guide for the British National Corpus version 1.0, 2000.

[3] Cristina Butnariu, Su Nam Kim, Preslav Nakov, Diarmuid Ó Séaghdha, Stan Szpakowicz, and Tony Veale. SemEval-2010 Task 9. The interpretation of noun compounds using paraphrasing verbs and prepositions. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, DEW '09, page 100–105, 2009.

[4] Pamela Downing. On the creation and use of English compound nouns. *Language*, 53(4):810–842, 1977.

[5] Dan Flickinger, Yi Zhang, and Valia Kordoni. DeepBank. A dynamically annotated treebank of the Wall Street Journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, page 85–96, Lisbon, Portugal, 2012. Edições Colibri.

[6] Roxana Girju, Dan Moldovan, Marta Tatu, and Daniel Antohe. On the semantics of noun compounds. *Computer Speech & Language*, 19(4):479–496, 2005.

[7] Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. SemEval-2007 Task 04. Classification of semantic relations between nominals. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, page 13–18, Prague, Czech Republic, 2007.

[8] Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and

Zdeněk Žabokrtský. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, page 3153–3160, Istanbul, Turkey, May 2012.

[9] Iris Hendrickx, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Stan Szpakowicz, and Tony Veale. SemEval-2013 Task 4. Free paraphrases of noun compounds. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, page 138–143, Atlanta, Georgia, USA, 2013.

[10] Mirella Lapata and Alex Lascarides. Detecting novel compounds. The role of distributional evidence. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, page 235–242, 2003.

[11] M. Lauer and M. Dras. A probabilistic model of compound nouns. In *Proceedings of the 7th Australian Joint Conference on AI*, 1994.

[12] Mark Lauer. *Designing Statistical Language Learners. Experiments on Noun Compounds*. Doctoral dissertation, Macquarie University, Sydney, Australia, 1995.

[13] Charles Na Li. *Semantics and the Structure of Compounds in Chinese*. PhD thesis, University of California, Berkeley, 1972.

[14] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpora of English. The Penn Treebank. *Computational Linguistics*, 19:313–330, 1993.

[15] Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. Annotating noun argument structure for NomBank. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, page 803–806, Lisbon, Portugal, 2004.

[16] Preslav Nakov. On the interpretation of noun compounds: Syntax, semantics, and entailment. *Natural Language Engineering*, 19(3):291–330, 2013.

[17] Diarmuid Ó Séaghdha. Learning compound noun semantics. Technical Report UCAM-CL-TR-735, University of Cambridge, Computer Laboratory, Cambridge, UK, 2008.

[18] Diarmuid Ó Séaghdha and Ann Copestake. Interpreting compound nouns with kernel methods. *Journal of Natural Language Engineering*, 19(3):331–356, 2013.

[19] Emily Pitler, Shane Bergsma, Dekang Lin, and Kenneth Church. Using web-scale n-grams to improve base NP parsing performance. In *Proceedings of the 23rd International Conference on Computational Linguistics*, page 886–894, Beijing, China, 2010.

[20] Tony Rose, Mark Stevenson, and Miles Whitehead. The Reuters Corpus Volume 1. From yesterday's news to tomorrow's language resources. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, volume 2, page 827–832, Las Palmas, Spain, 2002.

[21] Stephen Tratz and Eduard Hovy. A taxonomy, dataset, and classifier for automatic noun compound interpretation. In *Proceedings of the 48th Meeting of the Association for Computational Linguistics*, page 678–687, Uppsala, Sweden, 2010.

[22] David Vadas and James Curran. Adding Noun Phrase Structure to the Penn Treebank. In *Proceedings of the 45th Meeting of the Association for Computational Linguistics*, page 240–247, Prague, Czech Republic, 2007.

[23] David Vadas and James R Curran. Parsing noun phrases in the Penn Treebank. *Computational Linguistics*, 37(4):753–809, 2011.

The PropBank-Based
Persian Semantic Valency Lexicon

Azadeh Mirzaei[1] and Amirsaeid Moloodi[2]

[1]Department of Linguistics,
Allameh Tabataba'i University, Tehran, Iran,
[2]Department of Foreign Languages and Linguistics,
Shiraz University, Shiraz, Iran
E-mail: azadeh.mirzaei@atu.ac.ir, amirsaeid.moloodi@gmail.com

Abstract

This paper describes the procedure of developing the Persian semantic valency lexicon based on Persian Proposition Bank. This lexicon recorded about 14983 verb senses with their argument structures. For preparing the entries of this lexicon, a list of verbs was extracted from Persian semantically annotated corpus with about 30000 sentences, and the argument structure of each sentence was selected, corrected, completed and specified to represent the argument structures of the related extracted verbs. Afterwards amongst the sentences proposed by the annotation tool, one or two sentences were chosen as the example of each verb sense. Then a well-known definition was selected for distinct senses, and even if the verb had a metaphorical meaning, it was introduced as well. If the verb was polysemous, each definition and its related argument structure received a sense number. This lexical resource is an invaluable and useful data in linguistic studies, teaching the Persian language, and lexicography especially regarding the fact that prior to the current project, there was only a syntactic valency lexicon [12] with about 5000 entries in the Persian language and also there were no similar semantic instances.

1 Introduction[1]

FrameNet (Fillmore et al., [1], [2] and [3]) is based on a theory of meaning called Frame Semantics, according to Charles J. Fillmore ([4], [5] and [6]). In this approach, the meaning of a word is specified by the semantic frame assigned to it. The semantic frame includes an event as well as the relation, entity and the participants in it. Each member of the frame is known as a frame element (FE). All verbs evoking the same frame are called lexical units (LUs) of that one. The lexical units

[1] This paper was funded by Computer Research Center Of Islamic Science (CRCIS).

of each frame are verbs, but adjectives and nouns may also be included, especially in sentences with linking verbs (copula). VerbNet (Kipper et al., [7], [8], and [9]) groups verbs according to shared syntactic behaviors and also groups classes at a higher level according to shared semantic information; so in addition to defining the syntactic patterns characteristic of the verbs in each class, corresponding semantic representations which are showed with different semantic roles are appointed too. The present research explains how the Persian Semantic Valency Lexicon was developed based on the annotated Persian Proposition Bank. Although our lexicon resource used the tagsets of VerbNet to describe entries and their distinct senses, it discriminated between different senses of the verbs based on their propositional meaning which is the same as FrameNet. Prior to the present study, there was no VerbNet or FrameNet in Persian language.

2 Persian Proposition Bank

Persian Proposition Bank added a layer of predicate-argument information to the syntactic structures of Persian Dependency TreeBank [13], a manually syntactically annotated Persian corpus. This corpus comprises 29,982 sentences which were syntactically annotated according to dependency grammar, in which each word had one head and the head of the sentence, often the verb, was the dependent of an artificial root word (Kuebler et al. [10]).

In Persian Proposition Bank, verbs, predicative nouns, and predicative adjectives were chosen as heads in semantic relations, then argument and non-argument constituents related to each of them were assigned the proper semantic labels. The semantic labels were presented in two different kinds of tagsets. One kind was the numbered arguments like PropBank and the other one was VerbNet-based thematic roles. Persian Proposition Bank (PerPB) was prepared with numbered arguments and the other version Persian semantic role labeling corpus was annotated with thematic role labeling [11] which is the basis for the process of verb extraction to provide the distinct verb entries and also a basis for the development of the Persian Semantic Valency Lexicon.

Table 1: Statistics about the frequency of words in the PerPB.

Number of Sentences	29982
Average Sentence Length	16.61
Number of Verbs	62889
Number of distinct Senses	>9200
Number of distinct propositional Nouns	1300
Number of distinct propositional Adjectives	300

3 Semantic roles

In order to specify the argument structure of the verbs, 24 semantic roles were considered according to VerbNet [9]. The list of the arguments is presented in the table below. It should be noted that there are some small differences between this list and the list of semantic roles of VerbNet such as Theme1[2] and Theme2[3] .

Table 2: The list of the arguments

Agent	Initial-Location
Cause	Initial-Time
Experiencer	Final-Time
Patient	Possessor
Theme1	Possessum
Theme2	Stimulus
Destination	Extent
Location	Topic
Recipient	Attribute
Beneficiary	Co-agent
Source	Co-patient
Goal	Co-Theme

4 Annotation procedure

In Persian semantic role labeling corpus, each sentence was annotated according to the semantic role labeling guideline for verbal, nominal, and adjectival predicates such that for each sentence, predicative heads were first specified, and their argument structures in the sentence were specified afterwards. In addition to specifying argument structures, functional elements including adverbs of time, place, cause, goal and on the like were specified and annotated.

As we know some verb arguments can be omitted in the sentences. In such cases, the argument may not appear in the formal linguistic representation, although it is understood to be involved in the argument structure of the predicate.

1. ʔu hæme tʃiz ɹa goft
he/she every thing accusative marker said
She/he said the whole thing.

The predicate of sentence 1 is goft “say” whose argument structure is |Agent, Topic, Recipient|. However in this sentence one of the arguments of the verb namely the recipient of the message has been omitted. Thus, if we only extract the verbs and

[2] The participant is assigned Theme1 when the verb of the sentence just says something about its location.

[3] The participant is assigned Theme2 when it is central to an event and not structurally changed by it.

their argument structures from the semantic role labeling corpus, the resulting list will not necessarily show the complete and canonical argument structures of the verbs.

On the other hand, many verbs may be realized in their causative or passive forms. In such cases, the argument structure of the verb is not complete too.

2. a. I heated the chemicals to 200 degree Celsius.
   b. The chemicals were heated to 200 degree Celsius.

If we decide to specify the argument structure of "heat" based on 2.b, we will miss the verb's actor. Also, in causative sentences, the argument structure of the verb realized in the sentence is different from its non-causative form.

3. a. Her children do their homework.
   b. She made her children do their homework.

In 3b the presented argument structure of "do" as a causative verb includes |Cause, Actor, Theme|, whereas it has a different argument structure in non-causative situation.

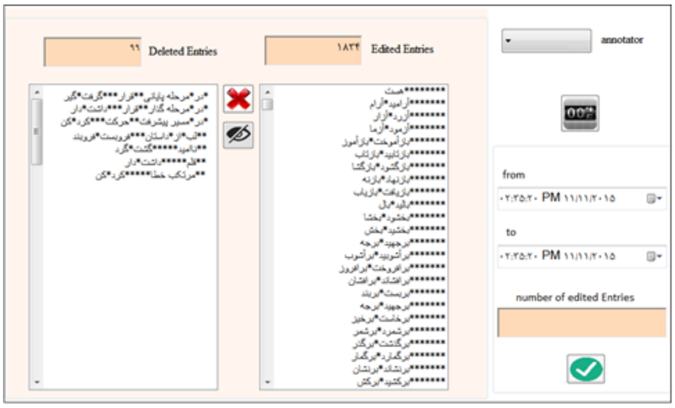
As a result, the verbs of the Persian semantic role labeling corpus were extracted with their argument structures in order to develop the Persian Semantic Valency Lexicon. The procedure of verb and argument structure extraction was as follows: each verbal argument structure was reported once in case the verb presented an identical argument structure. For instance if there were 150 cases of the verb "goft" with the same argument structure |Agent, Topic, Recipient| and similar or different inflections, this single argument structure was reported only once. If it was repeated 100 times with the argument structure |Agent, Topic|, it was extracted once more and finally for the entry "goft" all different realizations of the argument structures observed in the whole corpus were reported.

It is evident that there were many different argument structures in Persian semantic role labeling corpus which were all recorded for each verbal entry. The reasons of this variety of argument structures for an entry are as follows: There were some errors in annotation process; so the wrong structures would be deleted from the list of the valencies or edited for that entry.

As another reason, one can consider the omitted arguments of the verbs. As we know, Persian is a pro-drop language and because of the strong agreement between a verb and the person and number of its subject, the subject can be omitted which results in argument omitting. Also some of the arguments of the verbs are optional resulting in incomplete argument structures. In order to conquer this problem, the argument structures were compared with each other and if there was a full version, it was selected and if there was no full version, the argument structures of the verbs with the same sense were unified to represent their full version.

The other reason was the existence of different senses for an entry. In other words, if a single verbal form had different senses, it would be assigned different argument structures naturally. In such cases, the respective argument structures were numbered according to the frequency of each sense in the corpus so that sense number 1 would be considered the most frequent sense of the entry. To accomplish the task, we needed an annotation tool to represent the extracted verbal database from the corpus with some specified annotation facilities. The tool consisted of a search, an annotation and an administration panel.

In the administration panel (Figure 1) the admin (first writer of this paper) could supervise the annotators' activities, revise them and if it was necessary restore their deleted verb entries to the verbal database.



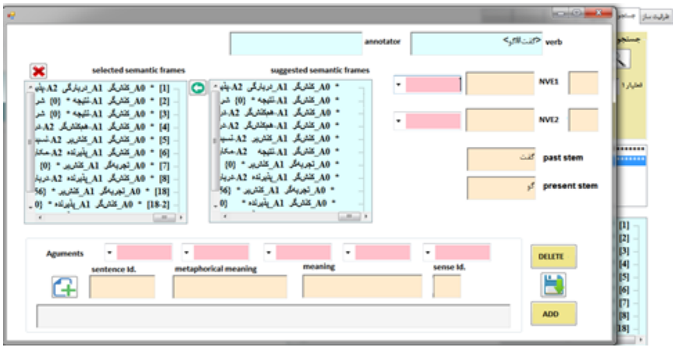
Deleted Entries ۹۹

Edited Entries ۱۸۳۴

from ۰۲:۳۵:۲۰ PM ۱۱/۱۱/۲۰۱۵

to ۰۲:۳۵:۲۰ PM ۱۱/۱۱/۲۰۱۵

Figure 1: administration panel



verb <گفت الآگو>

past stem گفت

present stem گو

Figure 2: annotation panel

Figure 2 shows the annotation panel. This panel would show all of the dissimilar valencies of each entry which were reported in the corpus and the full version of each sense was selected from among them, then it was numbered and recorded. As the size of the corpus was limited and some of the senses were missed, it was necessary to add the uncovered senses. Some of them were added based on the annotator's intuition while the others were found through googling. Also, the existing Persian thesauruses and lexicons helped us to find the other category of them. For these new senses the tool enabled the annotator to add them to the list of the entry senses with their corresponding arguments. Even the tool was equipped to add some new verbs missed in the corpus.

Alongside adding new senses, the program showed the annotator the example sentence from which the argument structure had been extracted. These example sentences were useful for disambiguating the senses and also the annotator could select the most appropriate sentence representing that specific sense. When there was no clear example it was possible to add a new one which was not artifact and was found by googling. Another feature of our annotation panel was assigning a definition to all of the selected senses. If the entry itself was the prototype of the definition, it itself was entered as the definition such as "say" which was recorded as the definition of "express", "assert" and "state". Otherwise the first definition which rushed through the annotator's mind would be selected as the definition of the sense.

As in the recent project the list of passive and causative forms of entries were limited and their exclusion wouldn't interrupt us to achieve the basis for our lexicon so just those categories of the entries which were regular and could be produced based on the rules were removed while the irregular ones with different lexical form in the passive or causative state were reserved. In the future work we will add all of the regular and irregular forms as a feature of its entry and our lexicon will be enriched.

Also it was possible to modify the entries. For example, in the case of passive forms, the user could convert the entry into the active form or in the case of wrong entry it was possible to correct it. In such cases, the program searched the database and if the entry was new, it was included and if it already existed, the program merged the two entries while checking the argument structures, deleting repeated ones and presenting a combination of valencies of both. It would then be recorded as just one entry in the list of the verbs.

For Persian Proposition Bank, there was inter-annotator agreement study and the corpus was evaluated with Kappa statistics measurement. The agreement on the role identification and the role classification was very high for this corpus which was the basis of Persian Semantic Valency Lexicon. Additionally the annotation process was controlled and checked by an adjudicator (the first writer of this paper) to make sure the annotations were consistent and correct.

5 Annotators

The annotators consisted of 4 PhD candidates (linguistics), and 2 MA graduates (1 linguistics graduate, 1 Persian language and literature graduate) who were native Persian speakers. Annotators were presented and trained with a comprehensive guidelines describing all the semantic roles with abundant examples.

6 Statistics

Table 3 shows the important statistics of our lexicon resource.

Table 3: Statistics about the frequency of verbs in the Persian Semantic Valency Lexicon

The number of distinct verb entries	9435
The number of distinct senses	14983
The number of simple and derivational verbs as a distinct entry	465
The number of simple and derivational verbs as a distinct sense	1828
The number of complex verb entry (compound, incorporating, metaphorical etc.)	8970
The number of complex verb as a distinct sense	13155

7 Conclusion

In this study first of all the list of the verbs in Persian Semantic Role Labeling Corpus was extracted and used as the entry of our lexicon. Then based on the unique argument structures, the distinct verb senses of each entry were reported and numbered according to their frequencies. These verb senses also included the metaphorical forms of the entry and the augmented ones added through googling or the use of thesauruses. This process resulted in 9435 distinct verb entries and 14983 unique senses. Each sense had a specific definition and at least one example sentence. In the future work the different argument structures with the same sense can be processed and the parts which can be omitted are tagged as a useful feature of the full version. Also the passive and causative forms of each entry would be included and all the verb senses would be clustered based on their shared syntactic and semantic behaviors. This dataset and the annotation tool would be presented for research purposes.[4]

8 Acknowledgements

The project was funded by Computer Research Center of Islamic Sciences (CRCIS). We really appreciate the linguists who helped us in annotating: Parinaz Dadras, Saeedeh Ghadrdoost-Nakhchi, Manoucher Kouhestani, Mostafa Mahdavi, Neda Poormorteza-Khameneh, and Salimeh Zamani; and the programmer who helped us in the process of the development of the lexicon: Fatemeh Sedghi; and other colleagues especially Mahdi Behniafar.

References

[1] Fillmore, Charles J. and BT Sue Atkins (1998) FrameNet and lexicographic relevance. In Proceedings of the First International Conference on Language Resources and Evaluation, pp. 28-30. Granada, Spain.

[4] http://www.peykaregan.com/fa/production-show-station-view

[2] Fillmore, Charles J. and Collin F. Baker (2001) Frame semantics for text understanding. In Proceedings of WordNet and Other Lexical Resources Workshop.

[3] Fillmore, Charles J., Christopher R. Johnson, and Miriam RL Petruck (2003) Background to framenet. International journal of lexicography 16.3: 235-250.

[4] Fillmore, Charles J. (1976) Frame semantics and the nature of language. In Annals of the New York Academy of Sciences, Vol. 280, pp. 20-32.

[5] Fillmore, Charles J. (1977) The case for case reopened. Syntax and semantics 8.1977: 59-82.

[6] Fillmore, Charles (1982) Frame semantics. Linguistics in the morning calm. 111-137.

[7] Kipper, Karin, Martha Palmer, and Owen Rambow (2002) Extending propbank with verbnet semantic predicates. In Proceedings of the 5th Conference of the Association for Machine Translation in the Americas (AMTA-2002), Tiburon, CA, USA, October (pp. 6-12).

[8] Kipper, Karin, Anna Korhonen, Neville Ryant and Martha Palmer (2006) Extending VerbNet with novel verb classes. (2006) Proceedings of LREC. Vol. 2006. No. 2.2..

[9]] Kipper, Karin, Anna Korhonen, Neville Ryant and Martha Palmer. (2008) A large-scale classification of English verbs. Language Resources and Evaluation 42, no. 1: 21-40.

[10] Kuebler, Sandra, Ryan McDonald, and Joakim Nivre (2009) Dependency Parsing. Synthesis Lectures on Human Language Technologies. Morgan and Claypool Publishers.

[11] Mirzaei, Azadeh and Amirsaeid Moloodi (2015) First Persian Semantic Role Labeling Corpus. Language Science, Volume 3, Issue 3.

[12] Rasooli, Mohammad Sadegh, Amirsaeid Moloodi, Manouchehr Kouhestani, and Behrouz Minaei-Bidgoli (2011) A syntactic valency lexicon for Persian verbs: The first steps towards Persian dependency treebank. In 5th Language and Technology Conference (LTC): Human Language Technologies as a Challenge for Computer Science and Linguistics (pp. 227-231).

[13] Rasooli, Mohammad Sadegh, Manouchehr Kouhestani and Amirsaeid Moloodi (2013) Development of a Persian syntactic dependency treebank. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 306-314).

Signals of Attribution in the Prague Dependency Treebank

Lucie Poláková, Pavlína Jínová and Jiří Mírovský

Faculty of Mathematics and Physics
Charles University in Prague
E-mail: {polakova|jinova|mirovsky}@ufal.mff.cuni.cz

Abstract

The paper aims at mining a richly annotated treebank for features relevant in automatic annotation/detection of attribution – ascription of text contents to agents who expressed them. We find three such features, implement an automatic procedure to detect attribution relations in our data and evaluate its results.

1 Introduction

In discourse-oriented linguistic research, attribution, or the ascription of text contents to the agents (sources) who expressed them, has become an important component of analysis, e.g. in the Penn Discourse Treebank [8], or it even developed to independent annotation projects, cf. Pareti [6].

Attribution relations (ARs) can be signaled with a range of language means. Mostly, it is clauses containing verbs of saying and thinking, but also further, non-verbal attribution phrases, compare Example 1 with two contents attributed to somebody else than the author. The example contains a prepositional signal *according to Kalina* and a clausal (verbal) signal *he remarks*.[1]

(1) A special category is the bank's award for the best Czech recording. **According to Kalina**, *this is an insurance for the case that the domestic production fails in all other categories.* However, *that did not happen*, **he remarks**.

In the Prague Dependency Treebank (PDT, Czech journalistic texts, [2]), annotation of discourse relations was first introduced in 2013 [7] but with no annotation of attribution so far. Before that, a complex manual analysis on three levels of description (morphology, surface and underlying syntax = tectogrammatics, [5])

[1] Attributed contents in Example 1 are highlighted in italics, attribution cues in bold.

had been carried out. Some of these annotated features appeared to be of great advantage for annotating intra-sentential discourse information.

The aim of this paper is twofold: i) to detect which attributes from the rich PDT annotation (or their combination) capture signals of attribution relations, and ii) to evaluate the reliability of these signals for an automatic annotation of this phenomenon. This is quite a natural next step towards a complex description of discourse relations. We are aware that the given task is partly dependent only on semantic and pragmatic features and as such cannot be fully automatized, our goal is therefore rather in determining how far we can facilitate the task by relying on already available information.

2 Preparatory Analysis

2.1 Method

Similarly as Pareti [6], we recognize an attribution relation as consisting of three main elements: the source of the attribution (agent expressing the contents), the attributed content and the cue – typically an attribution verb, less frequently also prepositions, adverbs, punctuation marks etc.

The research in this paper is targeted for future assignment of attribution primarily to discourse arguments and relations, thus it addresses mainly the possibilities of the identification of the cue.[2] Also, verbal cues that only introduce a sentence constituent, as in *He announced the break of contacts with the rebels.* are not targeted here, as non-clausal sentence constituents alone are not annotated as discourse arguments in the PDT so far.

To obtain a view of possible signals of attribution in the PDT, a manual inspection or random data samples was conducted, resulting in a list of signals which was afterwards further analyzed. Basically, morphological, syntactic and lexical features were encountered besides features connected with the text structure. Five attributes of the tectogrammatical layer seemed to represent the core of attribution signals; for each of them, 50 random occurrences in the corpus were examined to estimate their reliability for an automatic annotation. Three most promising attributes from these five are described in detail in Section 2.2 below. Other signals, assessed as either less distinctive or too rare for our purposes, are not further addressed in this paper.

2.2 Tracked Signals: Reported Speech and Verbs of Saying

Reported speech in the tectogrammatical representation is marked with the attribute *is_dsp_root* – **the root of a direct speech**. The goal of introducing this attribute was originally to mark syntactically unanchored reported contents (i.e. a reported

[2] So far, it does not concern the identification of sources, and only partly investigates the attributed contents, although the analyzed attributes in the PDT mostly also directly point to these two attribution elements.

speech not representing an obligatory modification of a governing verb of saying). The attribute is nevertheless assigned also to syntactically incorporated reported contents, both those graphically signaled by quotation marks and those without them.

However, in some cases, the *is_dsp_root* attribute is marked inconsistently, as it was not the main focus of the tectogrammatical annotation. That is where **valency frames** (syntactico-semantic roles) of the verbs can significantly help. In the valency lexicon of Czech verbs Vallex ([3], [4]), whose electronic version can be linked to the verbs in the corpus (see below in 3.1), each verb belongs to a certain semantic class (like motion, perception, change). For our experiment, we selected the semantic class of communication, as it intersects the best with verbs of saying. Verbs of thinking (the class of mental action in Vallex) were left out from the experiment in this phase. The list of verbs of communication (in Vallex 2.7) comprises 431 verbal frames,[3] 391 of which are relevant for the analysis of attribution. The combination of a unique verbal frame ID and a desired valency frame constellation is a promising way to detect both attribution cues and contents. Also, in this way, irrelevant meanings of polysemous verbs can be sorted out, as they have different valency frames.

Syntactically unanchored reported speech appears in Czech typically in cases where an introductory verb does not open a valency position for the content of saying (no direct object possible) or the position of a direct object is taken by another expression, cf. the expression *utkání* [*match*] in Example 2. Such structures in the PDT annotation are interpreted as if a verb of saying was missing. It is therefore represented by **a newly established node with the *t-lemma* substitute *#EmpVerb*** (empty verb) in the position of a non-obligatory verb complement [5, p. 421ff].[4] In Example 2, the whole reported content *I managed to win important rallies, Hyo arranged for the mistakes* is rooted in a generated *#EmpVerb* node representing approximately the (missing) verb *saying*. At the same time, this empty verb node is in the position of verbal complement (the COMPL functor) with dual dependency both on the verb *zhodnotila* [*evaluated*] and the noun *Novotná*.

(2) *Dařilo se mi vyhrávat důležité výměny, o chyby se postarala Hyová, zhodnotila ani ne hodinové utkání Novotná.*

[*I managed to win important rallies, Hyo arranged for the mistakes, Novotná evaluated the not even one hour lasting match.*]

It can be considered a reliable signal for attribution, with the added value of directly pointing at the source – it is always the entity in the position of the secondary parent of the complement (*Novotná* in Example 2).

[3] one verb lemma can have several different frames

[4] referred to as #EmpVerb.COMPL in Table 1 below

3 Automatic Detection of Attribution

3.1 Experiment Setting

For testing the theoretical analysis from the previous section, we have implemented an automatic procedure for detection of ARs in the PDT data. The selected features are, again:

- is_dsp_root – reported speech signaled by a dedicated attribute
- Vallex – usage of one of selected verbs of saying, extracted from the semantic class of communication from the valency lexicon Vallex
- #EmpVerb.COMPL – syntactically unanchored direct speech represented by a generated empty verb node in the position of verbal complement

To detect verbs of saying, we used the annotation of semantic classes in the valency lexicon Vallex, as described above in Section 2.2. However, information from Vallex about verb frames and their membership in the semantic class of communication could not be used directly. Verbs in the PDT data are not linked to Vallex but instead to so-called PDT-Vallex, where there is no annotation of semantic classes. Unfortunately, these two lexicons are not compatible in a straightforward way. For transforming the information about semantic classes from Vallex to PDT-Vallex, we used an automatic alignment of these two lexicons created by Bejček [1].

The automatic procedure for the attribution detection was tested on a selection of 15 manually evaluated documents from the PDT, comprising in total of 563 sentences. In an attempt to avoid documents with contents attributed only to the author of the text, the documents were selected based on different proportions of occurrences of the attribute *is_dsp_root*, three documents did not contain any occurrence of this attribute at all.

3.2 Results

Table 1 shows numbers of hits of individual or combined features of the automatic procedure in the manually evaluated data and, for comparison, also in 9/10 of the whole PDT data. A "hit" means a position in the data where the procedure detected one or more signals of attribution, that means, where it found at least one signal that the text span is attributed to some other source than the author. Sentences attributed only to the author of the text were ignored in the manual evaluation, or, in other words, a zero hit of the procedure in such a sentence did not count as a positive result. If there were several signals of attribution for the same text span (typically a clause), we count it as one hit in the respective row of the table. It means that, for example, in the manually evaluated data *is_dsp_root* was detected 68 times as the only signal of attribution and 48 times together with a verb of communication.

In the manually evaluated data, the automatic procedure correctly identified 137 out of 182 attribution relations, and incorrectly marked 3 relations. This means that the precision was 98%, recall 75%, and F1-measure 85%.

Feature(s)	In manual evaluation	In 9/10 of the PDT
is_dsp_root	68	1,693
Vallex + is_dsp_root	48	1,022
Vallex	10	1,324
#EmpVerb.COMPL	7	84
#EmpVerb.COMPL + is_dsp_root	5	71
Vallex + #EmpVerb.COMPL + is_dsp_root	1	16
Vallex + #EmpVerb.COMPL	1	10
total number of hits	140	4,220
total number of sentences	563	43,955

Table 1: Numbers of hits of individual features in the manually checked data and in 9/10 of the whole PDT data.

The high precision of the automatic procedure is an encouraging result and, considering that we have at this moment implemented only three signals of ARs, we consider the recall and the F1-measure figures also quite satisfactory.

3.3 Analysis of the Results

As Table 1 shows, the *is_dsp_root* attribute is the most reliable signal for identification of the reported contents among the implemented attributes. It correctly identified, as a single signal or in combination, 122 out of 182 ARs present in our data. This attribute moreover precisely delimits the reported content (the t-node with this attribute and its subtree) and points to the cue (if any present). Using valency frames from Vallex is more complicated due to its potential false positivity (see below). We were able to correctly detect 57 cue verbs in 182 ARs, however, it should be noted that not all ARs have a verbal cue. The effectiveness of this feature could be increased by finer rules regarding the individual frames. *#EmpVerb.COMPL* is a very precise signal of ARs, but, at the same time, it is quite rare. There are only 181 occurrences of these structures in the 9/10 of the PDT data. But, this signal is linguistically interesting in one respect – it can show which verbs outside the core of *verba dicendi* also can introduce attributed contents. We came across Czech verbs roughly corresponding to English *to join in, to conclude, to repeat, to give up, to praise, to react, to be delighted* and so on.

From the 45 undetected attribution relations, more than a half (25) were cases of a reported speech without any introductory verb. Such sentences mostly appear in a longer sequence of uninterrupted direct speech. The verb of saying is usually used only once for such a sequence. In 19 of these cases, attributing the content to somebody else than the author would be nevertheless possible by tracking the use of first person singular or plural (which is typical in our data – mostly news interviews). The remaining 6 cases could be identified as reported speech only thanks to thematic progressions and semantics.[5] Further, 9 undetected ARs were marked lexically with *podle* [*according to*] phrase, *prý* [*reportedly*], and *údajně* [*allegedly*]. In 4 cases, the procedure did not identify a verb of saying because its valency frame did not match any frame in our Vallex-originated list. In the remaining cases, the content of saying was expressed only through a demonstrative pronoun, and so the verbal cue and the content appeared in different sentences. Finally, the procedure so far failed to recognize parenthetical attributive structures with reverse syntactic order of the type *as he claims*.

There were three false positive hits in the manually evaluated texts. Although this is a small number, the individual cases point at two systematic problems of the procedure. First, it is the identified verbs of saying uttered by the author himself about himself, including certain fixed connections like *lépe řečeno* [*or rather,* lit. *better said*]. Second, it is some non-speaking meanings of some polysemous verbs. Most of the irrelevant frames were sorted out by the semantic class in Vallex, but some can remain, cf. the meaning of the verb *potvrdily* [*confirmed*] in Example 3.

(3) *Vítkovice potvrdily výhrou 2:0 nad Uherským Brodem, že budou patřit k nejvážnějším kandidátům na postup.*

[*Vítkovice confirmed by winning 2:0 over Uherský Brod that it will belong to the most serious candidates for the advance.*]

4 Discussion and Conclusions

Despite the complexity of detecting ARs in a text, we believe to have shown with our experiment that this task can be significantly facilitated if reliable syntactic annotation is at one's disposal. A crucial role also plays an available electronic lexicon of verbs with their syntactico-semantic roles (valency lexicon). Being that far, only implementation of three strongest features suffices to achieve very high precision and a fair recall. The procedure can be easily enhanced by adding further, rather primitive features like switching the category of person, lexical cues (according to + proper names, allegedly) etc. The proposed procedure is useful for any Czech treebank with tectogrammatical analysis (with a necessary decrease in performance in case of solely automatic parsing). On the other hand, the use of the valency lexicon makes it language-dependent. Also, for the time being, the analysis and the automatic procedure does not concern verbs of thinking that are, in our opinion, even trickier in expressing attribution relations than verbs of saying. We plan to address this issue in future experiments. For our research, which focuses on assigning attribution to already annotated discourse relations and arguments, the proposed experiment is a promising start. Manual evaluation of the results revealed very well the nature of cases where the procedure fails, which is a valuable linguistic feedback for understanding attribution and its principles.

[5] There were also two cases in our sample data where it could not be decided at all to whom they should be attributed. These cases were excluded from the evaluation.

Acknowledgements

We gratefully acknowledge support from the Ministry of Education of the Czech Republic (projects LM2010013 (LINDAT/CLARIN) and LH14011) and the Czech Science Foundation (project P406/12/0658). This work has been using language resources developed, stored and distributed by the LINDAT/CLARIN project.

References

[1] Bejček E. (2015). *Automatické propojování lexikografických zdrojů a korpusových dat.* [*Automatic linking of lexicographic sources and corpus data.*] Ph.D. thesis, Charles University in Prague, Faculty of Mathematics and Physics.

[2] Bejček E., E. Hajičová, J. Hajič, P. Jínová, V. Kettnerová, V. Kolářová, M. Mikulová, J. Mírovský, A. Nedoluzhko, J. Panevová, L. Poláková, M. Ševčíková, J. Štěpánek and Š. Zikánová (2013). *Prague Dependency Treebank 3.0.* Data/software, Univerzita Karlova v Praze, MFF, ÚFAL, Prague.

[3] Lopatková M. (2008). Valence a její formální popis. Vybrané aspekty budování slovníku VALLEX. [Valency and Its Formal Description. Selected Aspects in Development of Valency Lexicon.] In: *Proceedings of Malý informatický seminář (MIS 2008).* Praha: Matfyzpress, pp. 58–88.

[4] Lopatková M., V. Kettnerová, E. Bejček, K. Skwarska and Z. Žabokrtský (2012). *VALLEX 2.6.* Data/software, ÚFAL MFF UK, http://ufal.mff.cuni.cz/vallex/2.6/. (The newest publicly available version is Vallex 2.7: http://ufal.mff.cuni.cz/vallex/2.7/.)

[5] Mikulová M., A Bémová, J. Hajič, E. Hajičová, J. Havelka, V. Kolářová, L. Kučová, M. Lopatková, P. Pajas, J. Panevová, M. Razímová, P. Sgall, J. Štěpánek, Z. Urešová, K. Veselá and Z. Žabokrtský (2006). *Annotation on the tectogrammatical layer in the Prague Dependency Treebank. Annotation manual*, Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Prague.

[6] Pareti, S. (2015). Annotating Attribution Relations across Languages and Genres. In: *Proceedings of the Eleventh Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, London, UK.

[7] Poláková L., J. Mírovský, A. Nedoluzhko, P. Jínová, Š. Zikánová and E. Hajičová (2013). Introducing the Prague Discourse Treebank 1.0. In *Proceedings of the 6th International Joint Conference on Natural Language Processing,* pp. 91—99, Nagoya, Japan.

[8] Prasad R., N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi and B. Webber (2008). The Penn Discourse Treebank 2.0. In *Proceedings of LREC 2008,* pp. 2961–2968, Marrakech, Morocco.

Adapting Constraint Grammar for Parsing Estonian Chatroom Texts

Dage Särg

Institute of Estonian and General Linguistics
University of Tartu
E-mail: dage@ut.ee

Abstract

The aim of the paper is to describe adapting Estonian Constraint Grammar rule set to a new language variety - chatroom texts. To do this, a chatroom corpus was first parsed with the rule set developed for standard written Estonian, and manually revised in order to get the baseline results. After that, parsing errors were analysed and the rules regarding clause boundaries, shallow syntactic analysis and dependencies were adapted for the given language variety. Finally, the results were calculated on a test corpus. All the indicators improved: the achieved precision and recall of syntactic tags were 85.16% and 93.35% respectively, labeled and unlabeled attachment scores were 84.60% and 82.15%.

1 Introduction

Syntactic parsing is one of the basic levels of natural language processing, and its correct output can be useful in many further tasks, e.g. machine translation or grammar correction. A Dependency Constraint Grammar [1], [2], i.e. a set of human-written rules which add or remove grammatical tags to tokens based on context, has been developed for standard written Estonian in order to perform this task [3].

However, a lot of language that people produce every day does not comply with the grammar rules that define a standard written language. Therefore, it is important to study different varieties of language, and to adapt the language processing tools for those varieties. Recently, a trend of processing non-canonical language has emerged: numerous papers have been published on automatic processing of different language varieties, e.g. spontaneous speech [4], clinical texts [5], Twitter tweets [6], etc. In addition, there have been workshops dedicated to non-canonical language, e.g. SANCL 2012 (Syntactic Analysis of Non-Canonical Language)[1].

[1] https://sites.google.com/site/sancl2012/home

This paper also follows this trend: its goal is to give an overview of an attempt to adapt the Estonian CG rule set for chatroom language. The CG rule set has been previously adapted for shallow parsing of transcribed Estonian speech [7], [8] and dialects [9], but dependency parsing has only been performed on standard written Estonian.

The paper is organised as follows. Section 2 describes the corpus that was used for the study. In Section 3, baseline results are presented and most common parsing errors are discussed. After that, Section 4 gives an overview of adapting the rules for chatroom language, and in Section 5, final results are presented and discussed.

2 Data

The corpus that was used for this study is a subset of chatroom texts of the Estonian New Media Corpus[2]. It contains 10 text fragments from 9 different chatrooms. Each fragment is approximately 2000 tokens long (excluding punctuation) and the total size of the corpus is 19,809 tokens. Sentences are not specifically annotated in the corpus, instead, the text that one user enters at a time is regarded as a sentence. In total, the corpus consists of 5,204 such sentences. Therefore, it can be seen that the average sentence length in chatrooms is only 4 words which is considerably smaller than in the Estonian Dependency Treebank where the average sentence length is 14 words[3].

Eight of the nine chatrooms in the subset do not have a specific topic and the chats feature mostly everyday life of young people: school, work, relationships, free time, etc. The chat in the ninth chatroom is focused on technology, however, everyday life topics are discussed there as well.

The number of users in chatrooms varies greatly: during those 2000-token-fragments, there are 8-72 users who contribute at least one line of text to the chat. However, many of those users do not take actively part in the chat but just greet the others when they have logged in.

3 Results with unadapted rule set

First, in order to find out how the syntax of chatroom language differs from standard written Estonian and what the main types of parsing errors would be, the whole corpus was parsed, using the VISL-CG3[4] parser and Estonian CG rule set. The quality of morphological analysis of chatroom texts is far from being perfect [11] and as CG rules are highly dependent on morphological information, it would have been a large source of errors in syntactic analysis. Therefore, the morphological analysis of the corpus had been manually corrected. Special syntactic tags

[2] http://www.cl.ut.ee/korpused/segakorpus/uusmeedia/

[3] https://www.keeletehnoloogia.ee/et/ekt-projektid/vahendid-teksti-mitmekihiliseks-margendamiseks-rakendatuna-koondkorpusele/soltuvussuntaktiliselt-analuusitud-korpus

[4] http://beta.visl.sdu.dk/cg3.html

for discourse particles and emoticons were added to the tag set, as well as basic dependency rules for those tags.

The parsed corpus was manually revised and the initial results are presented in Figure 1 together with the results that have been achieved by [8], [9], and [10] on other varieties of Estonian. In chatrooms, 84.39% of the tokens had been analysed unambiguously, the most frequent type of ambiguity was between an adverbial and a postmodifier.

As can be seen from Figure 1, the precision and recall of syntactic tags in chatrooms are considerably lower than those achieved for speech and dialects (the low results of standard written Estonian are due to the fact that the morphological analysis had not been manually corrected, unlike in other language varieties). In chatrooms, 8.01% of the tokens (excluding punctuation) had been assigned an incorrect syntactic tag, most error-prone syntactic functions were subject (32% of all the errors), predicative (22%), and adverbial (10%). Almost half of the subjects with an incorrect syntactic function were actually direct addresses which are classified as subjects according to the CG rules of standard written Estonian. Predicatives had been assigned an incorrect tag mostly in elliptical sentences where a predicate had been left out. As this cannot be done in standard Estonian, there were no rules in the rule set for these kinds of sentences.

Labeled attachment score (LAS) and unlabeled attachment score (UAS) of the corpus were also considerably lower than those that have been achieved for literary language: LAS was 71.86% and UAS was 74.95% in chatrooms. This means that more than a quarter of all tokens were assigned an incorrect head or were not assigned a head at all. The tokens that received an incorrect head were mostly discourse particles (24.69%) and adverbials (24.63%), but also subjects (13.62%) and finite predicates (8.26%). Most of the tokens that were not assigned a head at all were discourse particles (64%) and emoticons (17%). A third (33%) of tokens with an incorrect head had also been assigned an incorrect syntactic tag.

	Chatrooms	**Speech[8]**	**Dialects[9]**	**Standard[10]**
Precision	83.97	90.2	87-89	72.0
Recall	91.99	97.6	96-97	92.6
UAS	74.95	NA	NA	83.4
LAS	71.86	NA	NA	80.3

Figure 1: Baseline results (%) of parsing chatroom language compared to the indicators achieved on other varieties of Estonian.

4 Adapting the rules

4.1 Clause boundaries

In order to adapt the rules, a 5,909-word subcorpus containing fragments from three chatrooms was used as a development corpus. The first stage in adapting the rules was to modify clause boundary detection rules. Clause boundary detection rules of standard written Estonian rely strongly on punctuation, but in chatrooms, punctuation is often not used or is used incorrectly. Therefore, some rules needed to be added, taking into account specific words (conjunctions, relative pronouns) that often mark clause boundaries, as well as the presence of several finite verb forms. Example 1 features the latter case: *on* and *ütlege* are both finite verbs, and in standard written Estonian they would be separated by a comma. For chatrooms, a rule was written that adds a clause boundary marker to a finite main verb if it is preceded by another finite main verb form.

(1)	palju	rahvast	siin	on	ütlege	JAA
	many	people-SG.PART	here	is	say-IMP.2PL	YES

'how many people are there say YES'

4.2 Syntactic functions

After that, the rules for assigning syntactic functions were taken into consideration. The rules needed to be modified mostly for treating elliptical sentences which are really prevalent in chatrooms and from which almost any syntactic function can be missing. An example of a parsed chatroom sentence before and after adapting the rules is presented on Figure 2. The sentence contains neither a predicate nor a subject, and therefore, the predicative *head* should be the head and the adverbials its dependents. However, as the rules in the standard written language rule set do not allow sentences where there is a predicative but no predicate, it is initially incorrectly tagged as an adverbial. Adding a rule that tags an adjective in nominative case as a predicative when there is no predicate in the clause and the adjective is not followed by a noun in nominative case helps to solve this problem.

As direct addresses are widely used in chatrooms, it was decided that they should be separated from subjects, and therefore, a new tag was added to the tag set. In standard written Estonian, direct address is placed between commas, however, in chatrooms this cannot be used as punctuation is often missing. Therefore, while writing the new rules, only proper names (including usernames) in nominative case and a small list of words that mark belonging to some kind of group (men, people, etc.) were considered as possible direct addresses. An example of a sentence with a direct address is presented on Figure 3. The proper names (chatroom usernames) *Caspar* and *operaator-k6ps* are actually direct addresses, but with the rule set of standard written language, they get tagged as objects because the plural 2nd person verb form in the sentence would only allow 'you' as a subject. Adding a separate

```
"<üsna>"                                  "<üsna>"              'quite'
"üsna" L0 D <0> @ADVL #1->0               "üsna" L0 D <0> @ADVL #1->2
"<head>"                                  "<head>"              'good-pl.nom'
"hea" Ld A pos pl nom @ADVL #2->1         "hea" Ld A pos pl nom @PRD #2->0
"<juba>"                                  "<juba>"              'already'
"juba" L0 D @ADVL #3->4                   "juba" L0 D @ADVL #3->2
"<tegelt>"                                "<tegelt>"            'actually'
"tegelikult" L0 D @ADVL #4->1             "tegelikult" L0 D @ADVL #4->2
```

Figure 2: Parse tree of the sentence *üsna head juba tegelt* 'quite good already actually' before (left) and after (right) adapting the rules

```
"<tehke>"                                                   'do-IMP.2PL'
"tege" Lke V main imper pres ps2 pl ps af @FMV #1->0
"<siis>"                                                    'then'
      "siis" L0 D @ADVL #2->1
"<Caspar>"                                                  'Caspar'
      "Caspar" L0 S prop sg nom cap @OBJ #3->1
"<ja>"                                                      'and'
      "ja" L0 J crd @J #4->5
"<operaator-k6ps>"                                          'operaator-k6ps'
      "operaator-k6ps" L0 S prop sg nom @OBJ #5->3
```

Figure 3: Parse tree of the sentence *tehke siis Caspar ja operaator-k6ps* 'do it then, Caspar and operaator-k6ps' before adapting the rules: direct addresses *Caspar* and *operaator-k6ps* are tagged as objects

tag for direct address solves this problem without having to rewrite the rules for tagging subjects.

4.3 Dependencies

The final stage of adapting the rule set regards dependency rules. After adding a new syntactic tag for direct addresses, it was necessary to write dependency rules for it as well. Generally, the head of the whole clause is the head of the direct address: predicate, or in elliptical sentences, a subject. There were also cases where the whole sentence consisted only of a discourse particle and a direct address - in these cases the discourse particle was treated as a head as it carries more meaning. Emoticons were excluded from the dependency trees because their use appeared to be most similar to punctuation marks: they were mostly used at the end of the sentence. For other syntactic tags, new rules mostly had to be written for elliptical sentences.

5 Results and discussion

In total, about 100 rules were added to the rule set which consisted of ca. 2000 rules, and in addition, about 50 of the existing rules were modified. The adapted rule set was tested on a 5,821-word subcorpus which consisted of text fragments of three chatrooms that were not in the development corpus. The results can be seen from Figure 4.

	Unadapted	Adapted
Precision	83.97	85.16
Recall	91.99	93.55
UAS	75.03	84.60
LAS	72.21	82.19

Figure 4: Results (%) before and after adapting the CG rule set for chatroom language

After adapting the rules, the precision and recall of syntactic tags on test corpus increased to 85.16% and 93.35% respectively. Most errors came still from assigning wrong tags to subjects and predicatives, but the recalls had significantly improved: for subjects, from 83.07% (subjects and direct addresses together) to 91.18% for direct addresses and to 85.04% for subjects. The recall of predicatives increased from 49.49% to 70.92%.

The UAS and LAS improved as well, to 84.60% and 82.19% respectively. The initial numbers were low because many tokens were not assigned a head at all, and this problem was fixed with the added rules. Still, the amount of tokens with an incorrect head decreased as well, from 17% to 15%. Writing new dependency rules for discourse particles helped to increase UAS the most: the UAS for discourse particles increased from 44.71% to 88.15%. The UAS of predicatives improved significantly as well: from 72.96% to 83.67%.

Despite the sentences in chatrooms being generally short and simple, the achieved parsing results are still slightly lower than the ones achieved for transcribed speech in [7] or [8] or dialects in [9]. The main reason for this is the fact that chatroom language is so heterogeneous: some users try to stick to the grammar rules of standard written Estonian while others produce text just as they please. As a result, it is not always possible to define the CG rules that would work for all the cases.

In the future, it could be useful to try and combine data-driven approaches with rule-based parsing in order to achieve better results. This has already been tried on standard written Estonian [10] where combining MaltParser with CG parser improved the LAS up to 1.5%. In addition, the adapted rule set could be applied on a different domain of Internet language or any other type of spontaneous written language to see if and how well it would scale.

References

[1] Karlsson, Fred, Voutilainen, Atro, Heikkilä, Juha and Anttila, Arto (1995). Constraint Grammar: a Language Independent System for Parsing Unrestricted Text. Berlin and New York: Mouton de Gruyter.

[2] Bick, Eckhard and Tino Didriksen (2015). CG3 - Beyond Classical Constraint Grammar. In *Proceedings of the 20th Nordic Conference of Computational Linguistics NODALIDA 2015*, pp. 31–39

[3] Müürisep, Kaili, Puolakainen, Tiina, Muischnek, Kadri, Koit, Mare, Roosmaa, Tiit and Uibo, Heli (2003). A New Language for Constraint Grammar: Estonian. In *International Conference Recent Advances in Natural Language Processing. Proceedings.* Borovets, Bulgaria, 10-12 September 2003, pp. 304–310.

[4] Bechet, Frederic, Nasr, Alexis and Benoit Favre (2014). Adapting dependency parsing to spontaneous speech for open domain spoken language understanding. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 01/2014*

[5] Savkov, Alexandar, Carroll, John and Jackie Cassell (2014). Chunking Clinical Text Containing Non-Canonical Language. In *Proceedings of the 2014 Workshop on Biomedical Natural Language Processing (BioNLP 2014).* pp. 77–82

[6] Kong, Lingpeng, Schneider, Nathan, Swayamdipta, Swabha, Bhatia, Archna, Dyer, Chris and Smith, Noah A. (2014). A Dependency Parser for Tweets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*

[7] Müürisep, Kaili and Uibo, Heli (2005). Shallow Parsing of Spoken Estonian Using Constraint Grammar. In Peter Juel Henrichsen and Peter Rossen Skadhauge (eds.) *Proceedings of NODALIDA 2005 Special Session on Treebanks for Spoken Language and Discourse* ; Copenhagen Studies in Language 32. Samfundslitteratur, pp. 105–118.

[8] Müürisep, Kaili and Nigol, Helen (2007). Disfluency Detection and Parsing of Transcribed Speech of Estonian. In *Proceedings of 3rd Language & Technology Conference Human Language Technologies as a Challenge for Computer Science and Linguistics.* Poznan, Poland: Wydawnictwo Poznanskie, pp. 483–487

[9] Lindström, Liina and Müürisep, Kaili (2009). Parsing Corpus of Estonian Dialects. In *Proceedings of the NODALIDA 2009 workshop Constraint Grammar and robust parsing.*

[10] Muischnek, Kadri, Müürisep, Kaili and Puolakainen, Tiina (2014). Dependency Parsing of Estonian: Statistical and Rule-based Approaches. In *Human Language Technologies – The Baltic Perspective.* pp. 111–118

[11] Kaalep, Heiki-Jaan and Muischnek, Kadri (2011). Morphological analysis of a non-standard language variety. In *NODALIDA 2011 Conference Proceedings.* Eds. Bolette Sandford Pedersen, Gunta Nešpore and Inguna Skadina. pp. 130–137.