UDW 2019

Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)

Proceedings

29–30 August, 2019
held within the **SyntaxFest** 2019, 26–30 August
Paris, France

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

978-1-950737-66-6

Preface

The Third Workshop on Universal Dependencies was part of the first SyntaxFest, a grouping of four events, which took place in Paris, France, during the last week of August:

- the Fifth International Conference on Dependency Linguistics (Depling 2019)
- the First Workshop on Quantitative Syntax (Quasy)
- the 18th International Workshop on Treebanks and Linguistic Theories (TLT 2019)
- the Third Workshop on Universal Dependencies (UDW 2019)

The use of corpora for NLP and linguistics has only increased in recent years. In NLP, machine learning systems are by nature data-intensive, and in linguistics there is a renewed interest in the empirical validation of linguistic theory, particularly through corpus evidence. While the first statistical parsers have long been trained on the Penn treebank phrase structures, dependency treebanks, whether natively annotated with dependencies, or converted from phrase structures, have become more and more popular, as evidenced by the success of the Universal Dependency project, currently uniting 120 treebanks in 80 languages, annotated in the same dependency-based scheme. The availability of these resources has boosted empirical quantitative studies in syntax. It has also lead to a growing interest in theoretical questions around syntactic dependency, its history, its foundations, and the analyses of various constructions in dependency-based frameworks. Furthermore, the availability of large, multilingual annotated data sets, such as those provided by the Universal Dependencies project, has made cross-linguistic analysis possible to an extent that could only be dreamt of only a few years ago.

In this context it was natural to bring together TLT (Treebanks and Linguistic Theories), the historical conference on treebanks as linguistic resources, Depling (The international conference on Dependency Linguistics), the conference uniting research on models and theories around dependency representations, and UDW (Universal Dependency Workshop), the annual meeting of the UD project itself. Moreover, in order to create a point of contact with the large community working in quantitative linguistics it seemed expedient to create a workshop dedicated to quantitative syntactic measures on treebanks and raw corpora, which gave rise to Quasy, the first workshop on Quantitative Syntax. And this led us to the first SyntaxFest.

Because the potential audience and submissions to the four events were likely to have substantial overlap, we decided to have a single reviewing process for the whole SyntaxFest. Authors could choose to submit their paper to one or several of the four events, and in case of acceptance, the program co-chairs would decide which event to assign the accepted paper to.

This choice was found to be an appropriate one, as most submissions were submitted to several of the events. Indeed, there were 40 long paper submissions, with 14 papers submitted to Quasy, 31 to Depling, 13 to TLT and 16 to UDW. Among them, 28 were accepted (6 at Quasy, 10 at Depling, 6 at TLT, 6 at UDW). Note that due to multiple submissions, the acceptance rate is defined at the level of the whole SyntaxFest (around 70%). As far as short papers are concerned, 62 were submitted (24 to Quasy, 41 to Depling, 35 to TLT and 37 to UDW), and 41 were accepted (8 were presented at Quasy, 14 at Depling, 9 at TLT and 9 at UDW), leading to an acceptance rate for short papers of around 66%.

We are happy to announce that the first SyntaxFest has been a success, with over 110 registered participants, most of whom attended for the whole week.

SyntaxFest is the result of efforts from many people. Our sincere thanks go to the reviewers who thoroughly reviewed all the submissions to the conference and provided detailed comments and suggestions, thus ensuring the quality of the published papers.

We would also like to warmly extend our thanks to the five invited speakers,

- Ramon Ferrer i Cancho - Universitat Politècnica de Catalunya (UPC)
- Emmanuel Dupoux - ENS/CNRS/EHESS/INRIA/PSL Research University, Paris
- Barbara Plank - IT University of Copenhagen
- Paola Merlo - University of Geneva
- Adam Przepiórkowski - University of Warsaw / Polish Academy of Sciences / University of Oxford

We are grateful to the Université Sorbonne Nouvelle for generously making available the Amphithéâtre du Monde Anglophone, a very pleasant venue in the heart of Paris. We would like to thank the ACL SIGPARSE group for its endorsement and all the institutions who gave financial support for SyntaxFest:

- the "Laboratoire de Linguistique formelle" (Université Paris Diderot & CNRS)
- the "Laboratoire de Phonétique et Phonologie" (Université Sorbonne Nouvelle & CNRS)
- the Modyco laboratory (Université Paris Nanterre)
- the "École Doctorale Connaissance, Langage, Modélisation" (CLM) - ED 139
- the "Université Sorbonne Nouvelle"
- the "Université Paris Nanterre"
- the Empirical Foundations of Linguistics Labex (EFL)
- the ATALA association
- Google
- Inria and its Almanach team project.

Finally, we would like to express special thanks to the students who have been part of the local organizing committee. We warmly acknowledge the enthusiasm and community spirit of:
Danrun Cao, Université Paris Nanterre
Marine Courtin, Sorbonne Nouvelle
Chuanming Dong, Université Paris Nanterre
Yoann Dupont, Inria
Mohammed Galal, Sohag University
Gaël Guibon, Inria
Yixuan Li, Sorbonne Nouvelle
Lara Perinetti, Inria et Fortia Financial Solutions
Mathilde Regnault, Lattice and Inria
Pierre Rochet, Université Paris Nanterre

Chunxiao Yan, Université Paris Nanterre

Marie Candito, Kim Gerdes, Sylvain Kahane, Djamé Seddah (local organizers and co-chairs),
and Xinying Chen, Ramon Ferrer-i-Cancho, Alexandre Rademaker, Francis Tyers (co-chairs)

September 2019

Program co-chairs

The chairs for each event (and co-chairs for the single SyntaxFest reviewing process) are:

- Quasy:
  - Xinying Chen (Xi'an Jiaotong University / University of Ostrava)
  - Ramon Ferrer i Cancho (Universitat Politècnica de Catalunya)
- Depling:
  - Kim Gerdes (LPP, Sorbonne Nouvelle & CNRS / Almanach, INRIA)
  - Sylvain Kahane (Modyco, Paris Nanterre & CNRS)
- TLT:
  - Marie Candito (LLF, Paris Diderot & CNRS)
  - Djamé Seddah (Paris Sorbonne / Almanach, INRIA)
  - with the help of Stephan Oepen (University of Oslo, previous co-chair of TLT) and Kilian Evang (University of Düsseldorf, next co-chair of TLT)
- UDW:
  - Alexandre Rademaker (IBM Research, Brazil)
  - Francis Tyers (Indiana University and Higher School of Economics)
  - with the help of Teresa Lynn (ADAPT Centre, Dublin City University) and Arne Köhn (Saarland University)

Local organizing committee of the SyntaxFest

Marie Candito, Université Paris-Diderot (co-chair)
Kim Gerdes, Sorbonne Nouvelle (co-chair)
Sylvain Kahane, Université Paris Nanterre (co-chair)
Djamé Seddah, University Paris-Sorbonne (co-chair)
Danrun Cao, Université Paris Nanterre
Marine Courtin, Sorbonne Nouvelle
Chuanming Dong, Université Paris Nanterre
Yoann Dupont, Inria
Mohammed Galal, Sohag University
Gaël Guibon, Inria
Yixuan Li, Sorbonne Nouvelle
Lara Perinetti, Inria et Fortia Financial Solutions
Mathilde Regnault, Lattice and Inria
Pierre Rochet, Université Paris Nanterre
Chunxiao Yan, Université Paris Nanterre

Program committee for the whole SyntaxFest

Patricia Amaral (Indiana University Bloomington)
Miguel Ballesteros (IBM)
David Beck (University of Alberta)
Emily M. Bender (University of Washington)
Ann Bies (Linguistic Data Consortium, University of Pennsylvania)
Igor Boguslavsky (Universidad Politécnica de Madrid)
Bernd Bohnet (Google)
Cristina Bosco (University of Turin)
Gosse Bouma (Rijksuniversiteit Groningen)
Miriam Butt (University of Konstanz)
Radek Čech (University of Ostrava)
Giuseppe Giovanni Antonio Celano (University of Pavia)
Çagrı Çöltekin (University of Tuebingen)
Benoit Crabbé (Paris Diderot University)
Éric De La Clergerie (INRIA)
Miryam de Lhoneux (Uppsala University)
Marie-Catherine de Marneffe (The Ohio State University)
Valeria de Paiva (Samsung Research America and University of Birmingham)
Felice Dell'Orletta (Istituto di Linguistica Computazionale "Antonio Zampolli" - ILC CNR)
Kaja Dobrovoljc (Jožef Stefan Institute)
Leonel Figueiredo de Alencar (Universidade federal do Ceará)
Jennifer Foster (Dublin City University, Dublin 9, Ireland)
Richard Futrell (University of California, Irvine)
Filip Ginter (University of Turku)
Koldo Gojenola (University of the Basque Country UPV/EHU)
Kristina Gulordava (Universitat Pompeu Fabra)
Carlos Gómez-Rodríguez (Universidade da Coruña)
Memduh Gökirmak (Charles University, Prague)
Jan Hajič (Charles University, Prague)
Eva Hajičová (Charles University, Prague)
Barbora Hladká (Charles University, Prague)
Richard Hudson (University College London)
Leonid Iomdin (Institute for Information Transmission Problems, Russian Academy of Sciences)
Jingyang Jiang (Zhejiang University)
Sandra Kübler (Indiana University Bloomington)
François Lareau (OLST, Université de Montréal)
John Lee (City University of Hong Kong)
Nicholas Lester (University of Zurich)
Lori Levin (Carnegie Mellon University)
Haitao Liu (Zhejiang University)
Ján Mačutek (Comenius University, Bratislava, Slovakia)
Nicolas Mazziotta (Université)
Ryan Mcdonald (Google)
Alexander Mehler (Goethe-University Frankfurt am Main, Text Technology Group)

Wolfgang Menzel (Department of Informatik, Hamburg University)
Paola Merlo (University of Geneva)
Jasmina Milićević (Dalhousie University)
Simon Mille (Universitat Pompeu Fabra)
Simonetta Montemagni (ILC-CNR)
Jiří Mírovský (Charles University, Prague)
Alexis Nasr (Aix-Marseille Université)
Anat Ninio (The Hebrew University of Jerusalem)
Joakim Nivre (Uppsala University)
Pierre Nugues (Lund University, Department of Computer Science Lund, Sweden)
Kemal Oflazer (Carnegie Mellon University-Qatar)
Timothy Osborne (independent)
Petya Osenova (Sofia University and IICT-BAS)
Jarmila Panevová (Charles University, Prague)
Agnieszka Patejuk (Polish Academy of Sciences / University of Oxford)
Alain Polguère (Université de Lorraine)
Prokopis Prokopidis (Institute for Language and Speech Processing/Athena RC)
Ines Rehbein (Leibniz Science Campus)
Rudolf Rosa (Charles University, Prague)
Haruko Sanada (Rissho University)
Sebastian Schuster (Stanford University)
Maria Simi (Università di Pisa)
Reut Tsarfaty (Open University of Israel)
Zdenka Uresova (Charles University, Prague)
Giulia Venturi (ILC-CNR)
Veronika Vincze (Hungarian Academy of Sciences, Research Group on Articial Intelligence)
Relja Vulanovic (Kent State University at Stark)
Leo Wanner (ICREA and University Pompeu Fabra)
Michael White (The Ohio State University)
Chunshan Xu (Anhui Jianzhu University)
Zhao Yiyi (Communication University of China)
Amir Zeldes (Georgetown University)
Daniel Zeman (Univerzita Karlova)
Hongxin Zhang (Zhejiang University)
Heike Zinsmeister (University of Hamburg)
Robert Östling (Department of Linguistics, Stockholm University)
Lilja Øvrelid (University of Oslo)

Additional reviewers

James Barry
Ivan Vladimir Meza Ruiz
Rebecca Morris
Olga Sozinova
He Zhou

Table of Contents

Invited Talk

Friday 30th August 2019

Arguments and adjuncts

Adam Przepiórkowski
University of Warsaw / Polish Academy of Sciences / University of Oxford

Abstract

Linguists agree that the phrase “two hours” is an argument in “John only lost two hours” but an adjunct in “John only slept two hours”, and similarly for “well” in “John behaved well” (an argument) and “John played well” (an adjunct). While the argument/adjunct distinction is hardwired in major linguistic theories, Universal Dependencies eschews this dichotomy and replaces it with the core/non-core distinction. The aim of this talk is to add support to the UD approach by critically examinining the argument/adjunct distinction. I will suggest that not much progress has been made during the last 60 years, since Tesnière used three pairwise-incompatible criteria to distinguish arguments from adjuncts. This justifies doubts about the linguistic reality of this purported dichotomy. But – given that this distinction is built into the internal machinery and/or resulting representations of perhaps all popular linguistic theories – what would a linguistic theory not making such an argument–adjunct distinction look like? I will briefly sketch the main components of such an approach, based on ideas from diverse corners of linguistic and lexicographic theory and practice.

Short bio

Adam Przepiórkowski is a full professor at the University of Warsaw (Institute of Philosophy) and at the Polish Academy of Sciences (Institute of Computer Science). As a computational and corpus linguist, he has led NLP projects resulting in the development of various tools and resources for Polish, including the National Corpus of Polish and tools for its manual and automatic annotation, and has worked on topics ranging from deep and shallow syntactic parsing to corpus search engines and valency dictionaries. As a theoretical linguist, he has worked on the syntax and morphosyntax of Polish (within Head-driven Phrase Structure Grammar and within Lexical-Functional Grammar), on dependency representations of various syntactic phenomena (within Universal Dependencies), and on the semantics of negation, coordination and adverbial modifcation (at different periods, within Glue Semantics, Situation Semantics and Truthmaker Semantics). He is currently a visiting scholar at the University of Oxford.

Building a treebank for Occitan: what use for Romance UD corpora?

Aleksandra Miletic*, Myriam Bras*, Louise Esher*,
Jean Sibille*, Marianne Vergez-Couret**
* CLLE-ERSS (CNRS UMR 5263), University of Toulouse Jean Jaurès, France
firstname.lastname@univ-tlse2.fr
** FoReLLIS (EA 3816), University of Poitiers, France

Abstract

This paper describes the application of delexicalized cross-lingual parsing on Occitan with a view to building the first dependency treebank of this language. Occitan is a Romance language spoken in the south of France and in parts of Italy and Spain. It is a relatively low-resourced language and does not have a syntactically annotated corpus as of yet. In order to facilitate the manual annotation process, we train parsing models on the existing Romance corpora from the Universal Dependencies project and apply them to Occitan. Special attention is given to the effect of this cross-lingual annotation on the work of human annotators in terms of annotation speed and ease.

1 Introduction

Occitan is a Romance language spoken across the south of France and in several areas of Italy and Spain. Although it has no official status in France, it has been recognized – among other regional languages – as part of the cultural heritage of France by the constitutional amendment Article 75-1 published in 2008. Ever since, there have been more efforts to strengthen the preservation and the dissemination of the language through the creation of electronic resources. The most notable such project was RESTAURE (Bernhard et al., 2018), which resulted in the creation of an electronic lexicon (Vergez-Couret, 2016) and a POS tagged corpus (Bernhard et al., 2018) for Occitan. However, Occitan does not yet have a syntactically annotated corpus. This paper presents the first efforts towards the creation of such a resource.

It is well-known that manual annotation is time-consuming and costly. In order to facilitate and accelerate the work of human annotators, we implement direct delexicalized cross-lingual parsing in order to provide an initial syntactic annotation. This technique consists in training a parsing model on a delexicalized corpus of a source language and then using the model to process data in the target language. The training is typically only based on POS tags and morphosyntactic features, whereas lexical information (i.e. the information related to the token and the lemma) is ignored. Thus, the model is able to parse the target language even though no target language content was present in the training corpus.

In the past, delexicalized cross-lingual parsing was used with mixed results due to the divergent annotation schemes in different corpora (McDonald et al., 2011). The Universal Dependencies project (Nivre et al., 2016) offers a solution to this issue: version 2.3 comprises over 100 corpora in over 70 different languages[1], all annotated according to the same annotation scheme. The use of such harmonized annotations has lead to cross-lingual parsing results consistent with typological and genealogical relatedness of languages (McDonald et al., 2013). These corpora have since been successfully applied to delexicalized parsing of numerous language pairs (Lynn et al., 2014; Tiedemann, 2015; Duong et al., 2015).

Lexicalized cross-lingual parsing was also considered as a possible solution, but was rejected for two main reasons. Firstly, to the best of our knowledge, there are no parallel corpora of Occitan that could have been of immediate use for techniques such as annotation projection. Secondly, Occitan data could have been adapted to lexicalized parsing through different techniques such as machine translation or devoweling (Tiedemann, 2014; Rosa and Mareček, 2018), but the effort needed for such an approach is not negligible. As already stated above, the work presented here was conducted as part of a corpus-building project, with the primary goal of accelerating the manual annotation process. The methods used to facilitate

[1] https://universaldependencies.org/

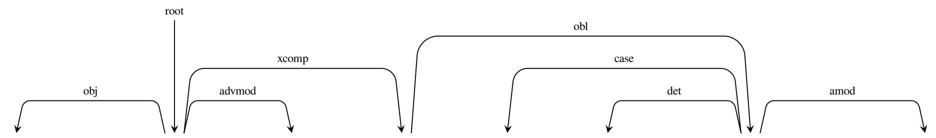
the annotation were therefore not to be more costly than manual annotation itself. Given this constraint, delexicalized cross-lingual parsing was chosen as the most straightforward approach.

Direct delexicalized cross-lingual parsing has been used to initiate the creation of an Old Occitan treebank. Scrivner and Kübler (2012) used Catalan and Old French corpora for cross-lingual transfer of both POS tagging and parsing. Unfortunately, we were unable to locate the resulting corpus. We therefore decided to implement delexicalized cross-lingual parsing based on the Romance corpora made available by the UD project. In this paper we present the quantitative evaluation of this process, but also the effects of this technique on the work of human annotators in terms of manual annotation speed and ease.

The remainder of this paper is organized as follows. First, we give a brief linguistic description of Occitan (Section 2); in Section 3 we describe the resources and tools used in our experiments; we then present the quantitative evaluation of the parsing transfer (Section 4) and analyze the impact of this method on the manual annotation (Section 5). Lastly, we draw our conclusions and discuss future work in Section 6.

2 Occitan

Occitan is a Romance language spoken in a large area in the south of France, in several valleys in Italy and in the Aran valley in Spain. It shares numerous linguistic properties with several other Romance languages: it displays number and gender inflection marks on all members of the NP, and it has tense, person and number inflection marks on finite verbs (cf. example 1). It is a pro-drop language with relatively free word order and as such it is closer to Catalan, Spanish and Italian than to French and other regional languages from the north of France.



(1)	Vos	vòli	pas	espaurugar	amb	lo	rescalfament	planetari
	you.ACC.PL	wanted.1SG	NEG	frighten	with	the.SG.M	warming	planetary.SG.M

'I didn't want to scare you with global warming.'

Another crucial property of Occitan from the NLP point of view is that it has not been standardized. It has numerous varieties organized in 6 dialectal groups (Auvernhàs, Gascon, Lengadocian, Lemosin, Provençau and Vivaro-Aupenc). Also, there is no global spelling standard, but rather two different norms, one called the *classical*, based on the Occitan troubadours' medieval spelling, and the other closer to the French language conventions (Sibille, 2000). This double diversity which manifests itself both on the lexical level and in the spelling makes Occitan particularly challenging for NLP.

To avoid the data sparsity issues that can arise in such a situation while working on small amounts of data, we decided to initiate the treebank building process with texts in Lengadocian and Gascon written using the classical spelling norm. Once we produce a training corpus sufficient to generate stable parsing models in these conditions, other varieties will be added.

3 Resources and tools

To implement cross-lingual delexicalized parsing, we used the Romance language corpora from the UD project as training material, we created a manually annotated sample of Occitan to be used as an evaluation corpus, and we used the Talismane NLP suite to execute all parsing experiments. Each of these elements is presented in detail below.

3.1 UD Romance corpora

Universal Dependencies v2.3 comprises 22 different corpora in 8 Romance languages (Catalan, French, Galician, Italian, Old French, Portuguese, Romanian, and Spanish). These corpora vary in size (from 23K tokens in the PUD corpora in French, Italian, Portuguese and Spanish to 573K tokens in the FTB corpus of French), as well as in terms of content: they include newspaper texts, literature, tweets, poetry, spoken language, scientific and legal texts.

Some of these corpora were excluded from our experiments. Some were eliminated based on the text genre. The Occitan corpus we are working on consists mainly in literary and newspaper texts. We therefore did not include corpora containing spoken language and tweets. Secondly, in order to ensure the quality of the parsing models trained on the corpora, we only selected those built through manual annotation or converted from such resources. Lastly, for practical reasons, we only kept the corpora that already had designated train and test sections. This resulted in a set of 14 corpora, but all 8 languages are represented (for the full list, see Section 4.1).

These corpora integrate different sets of morphosyntactic traits, and some of them implement a number of two-level syntactic labels. In order to maintain consistency between the training corpora, but also with the Occitan evaluation sample, no morphosyntactic traits were used in training, and syntactic annotation was reduced to the basic one-level labels.

3.2 Manually annotated evaluation sample in Occitan

In order to evaluate the suitability of the delexicalized models for the processing of our target language, we created an evaluation sample in Occitan. This sample contains around 1000 tokens from 4 newspaper texts, 3 of which are in Lengadocian and 1 in Gascon (cf. Table 1). The sample is tagged with UD POS tags, obtained by a conversion from an existing Occitan corpus which was tagged manually using EAGLES and GRACE tagging standards (Bernhard et al., 2018). As of yet, the sample contains no fine-grained morphosyntactic traits[2].

Sample	Dialect	No tokens	No POS	No labels
jornalet-atacs	Lengadocian	272	13	25
jornalet-festa	Lengadocian	353	13	24
jornalet-lei	Lengadocian	310	12	20
jornalet-estanguet	Gascon	217	12	24
TOTAL		1152	14	27

Table 1: Occitan evaluation sample

At the moment, the syntactic annotation is limited to first-level dependency labels (no complex syntactic labels). This is due to the fact that the annotation of this evaluation sample was in fact the first round of syntactic annotation in the project. It was therefore used to test and refine the general UD guidelines, but also to gather information as to which two-level labels may be necessary. The result of this analysis will be included in the next round of annotation.

The syntactic annotation of the sample was done manually using the brat annotation tool (Stenetorp et al., 2012). Each text was processed by one annotator who had extensive experience with dependency syntax, UD guidelines and the annotation interface (although not on Occitan), and one novice. The inter-annotator agreement on the sample in terms of Cohen's *kappa* (excluding punctuation marks) is 88.1. This can be considered as a solid result given that this was the very first cycle of annotation. All disagreements were resolved in an adjudication process, resulting in a gold-standard annotated sample.

3.3 Talismane NLP suite

For all parsing experiments described in this paper, we used Talismane (Urieli, 2013). It is a complete NLP pipeline capable of sentence segmentation, tokenization, POS tagging and dependency parsing. It currently integrates 3 algorithms: perceptron, MaxEnt, and SVM. The Talismane tagger has already been successfully used on Occitan for POS tagging in a previous project (Vergez-Couret and Urieli, 2015), on the outcomes of which the current project is founded. Talismane gives full access to the learning features, which can be defined by the user. Thus, it suffices to adapt the feature file in order to define the desired

[2] The original corpus annotation does encode some lexical traits, which will be recuperated and included in the UD conversion in immediate future. However, the original corpus does not contain any inflectional traits.

learning conditions: in our case, no lemma-based or token-based features were included in the feature set, which dispensed the user from the need to modify the learning corpora. This was particularly useful given the number of corpora used. However, numerous recent works have shown that tools based on neural networks outperform classical machine learning algorithms in tasks including dependency parsing, while often offering comparable practical advantages (Zeman et al., 2017; Zeman et al., 2018). One of the future steps in the continuation of this work will be to test neural network parsers on our data.

4 Transferring delexicalized parsing models to Occitan

We used Talismane's SVM algorithm to train models on the selected corpora. Learning was based on the POS tag features of the processed token and its linear and syntactic context, and different combinations thereof (34 features in total). Since the features were light, the training generated relatively compact models even for the largest corpora (the biggest at 130MB). The generated models were evaluated first on their respective test samples and then on the manually annotated Occitan sample. The results are discussed below.

4.1 Baseline evaluation

The goal of this first evaluation was to establish the baseline results for each model. This baseline was to be used to assess the stability of the models when transferred to Occitan. The results are given in Table 2. The corpus names contain the language code and the name of the corpus in lowercase. Parsing results are given as LAS[3] and UAS[4]. The top 5 models in terms of the LAS are highlighted in bold.

Corpus	Train size	Test size	LAS	UAS
ca_ancora	418K	58K	77.82	82.20
es_ancora	446K	52.8K	76.75	81.29
es_gsd	12.2K	13.5K	74.88	78.81
fr_partut	**25K**	**2.7K**	**82.41**	**84.60**
fr_gsd	364K	10.3K	78.51	81.81
fr_sequoia	52K	10.3K	78.29	80.71
fr_ftb	470K	79.6K	68.93	73.08
gl_treegal	16.7K	10.9K	73.91	78.79
it_isdt	**294K**	**11.1K**	**81.03**	**84.19**
it_partut	**52.4K**	**3.9K**	**82.66**	**85.22**
ofr_srcmf	136K	17.3K	69.41	79.09
pt_bosque	222K	10.9K	77.41	81.27
pt_gsd	**273K**	**33.6K**	**80.2**	**83.2**
ro_rrt	185K	16.3K	71.87	78.92
ro_nonstandard	155K	20.9K	65.59	75.45
es_ancora+gsd	458.2K	66.3K	73.14	78.24
fr_partut+gsd+sequoia	441K	23.3K	73.69	77.57
fr_partut+gsd+sequoia+ftb	911K	102.9K	74.87	78.55
it_isdt+partut	**346.4K**	**15K**	**81.78**	**84.66**
pt_bosque+gsd	495K	44.5K	76.09	81.47
ro_nonstand+rrt	340K	37.2K	67.21	76.06

Table 2: Baseline evaluation of models trained on UD Romance corpora

The LAS varies from 65.59 (ro_nonstandard) to 82.41 (fr_partut), and the UAS from 73.08 (fr_ftb) to 85.22 (it_partut), with the top 5 models acheiving an LAS > 80 and a UAS > 83. We also tested the option of merging several corpora in the same language (cf. the lower half of the table) under the supposition that,

[3] Labelled Attachment Score: percentage of tokens for which both the governor and the syntactic label were identified correctly.

[4] Unlabelled Attachment Score: percentage of tokens for which the governor was identified correctly, regardless of the syntactic label.

given the shared annotation scheme, this would equate to having a larger training corpus and boost the results. However, none of the combined corpora produced a model that surpassed the best performing individual model, although it_isdt+partut did score among the top 5. This seems to indicate that there are divergences in the application of the UD annotation scheme between different corpora of the same language, resulting in inconsistent annotations in the merged corpora. Indeed, at least one such discrepancy was spotted in the French corpora during this work: the temporal construction *il y a* 'ago' is annotated in three different ways in the GSD, ParTUT and Sequoia corpora. Nevertheless, it should be noted that such effects can also be due to the fact that the content of the combined corpora was simply concatenated and not reshuffled, which may have had a negative effect on the learning algorithm.

Nevertheless, since the baseline performances were not necessarily directly indicative of the results that each model would achieve on Occitan, all models generated in this step were tested on Occitan too.

4.2 Evaluation on Occitan

Table 3 details the results of the parsing evaluation on the manually annotated Occitan sample presented in section 3.2. The models are listed from best to worst in terms of LAS. Since the test sample contains around 1000 tokens, a different annotation of a single token constitutes roughly a 0.1% change in the parsing scores. Therefore, the scores are rounded to one decimal point.

Train corpus	LAS	UAS
it_isdt	71.6	76.0
it_isdt+partut	71.3	75.9
fr_partut+gsd+sequoia	70.8	75.7
fr_gsd	70.4	75.9
pt_bosque	70.0	75.3
it_partut	69.7	74.1
fr_partut+gsd+sequoia+ftb	69.6	74.4
fr_partut	69.4	74.6
es_ancora+gsd	69.1	74.9
es_ancora	69.0	75.3
gl_treegal	68.7	73.4
ca_ancora	68.6	75.2
fr_sequoia	68.6	73.3
es_gsd	67.8	73.4
fr_ftb	67.4	72.5
ro_rrt	67.1	72.2
ro_nonstand+rrt	66.6	72.0
pt_bosque+gsd	66.4	74.3
pt_gsd	63.1	73.3
ro_nonstand	60.2	72.7
ofr_scmrf	59.2	66.0

Table 3: Evaluation on the manually annotated Occitan sample. (Bold: models selected for further experiments.)

In this evaluation scenario, the LAS varies from 59.2 (ofr_scmrf) to 71.6 (it_isdt), whereas the UAS ranges from 66.0 (ofr_scmrf) to 76.0 (it_isdt). Rather surprisingly, among the top 5 models we find three based on French and Portugese corpora, although these languages are not traditionally considered as close to Occitan. What is more, the languages that have already been used for delexicalized parsing transfer on Occitan, namely Catalan and Old French (Scrivner and Kübler, 2012), come in as 14th and last, respectively. Also, the pt_bosque model scores here as 5th, whereas it was only 10th in the baseline evaluation. It is also interesting to note that the best results here come from large corpora, the smallest in the top 5 being pt_bosque with 222K tokens. Finally, the only model that did not suffer important performance loss is fr_partut+gsd+sequoia: it lost 2.9 LAS points and 1.9 UAS points, whereas the other four lost 7-10 LAS points and 6-8 UAS points. This may indicate that the diversity of linguistic content that was a disadvantage in the baseline evaluation actually provided robustness to the model which allowed it to maintain its performance when transferred to Occitan. This however has to be further investigated.

For the following step, we selected the best performing model for each of the languages in the top 5 (it_isdt, fr_partut+gsd+sequoia, pt_bosque) and used them to pre-annotate new Occitan samples. It is important to note that the difference in scores between it_isdt and it_idst+partut is explained by different annotation of 3 tokens when it comes to LAS, and 1 token when it comes to UAS, whereas the difference between it_isdt+partut and e.g. pt_bosque is much more important. However, we preferred having models based on different languages and comparing their performances rather than adhering strictly to the quantitative results.

5 Annotating Occitan: parsing process and manual correction analysis

The models selected in the previous step were applied to new samples of Occitan text. Coming from an existing corpus, these samples already had a manual POS annotation needed to put the delexicalized models to work. The resulting annotation was then submitted for validation to an experienced annotator. The corrected analysis was used as a gold standard against which the initial automatic annotation was evaluated. The manual annotation process also allowed us to observe the specificities of the annotation produced by the models and their impact on the manual annotation process.

5.1 Parsing new Occitan samples with selected UD models

Each of the 3 selected models was used to parse a new, syntactically unannotated sample of some 300 tokens of Occitan text. In order to minimize the bias related to the intrinsic difficulty of the text, we selected samples from the same source[5]. The annotation produced by the models was filtered: since it can be very time-consuming to correct erroneous dependencies, we only retained the dependencies for which the parser's decision probability score was >0.7. This was possible thanks to a Talismane option allowing to output the probability score for each parsing decision. Several other thresholds were tested (0.5, 0.6, 0.8, 0.9), and 0.7 was chosen for a balanced ratio between the confidence level and the sample coverage. Although research on parser confidence estimation has shown that more complex means may be needed to obtain reliable confidence estimates (Mejer and Crammer, 2012, e.g.), the Talismane probability scores have already been used in this fashion and have been judged as adequate by human annotators (Miletic, 2018).

Table 4 shows the size of each sample, the model used to parse it and the coverage of the sample by the model when the 0.7 probability filter is applied. This partial annotation was then imported into the brat annotation tool and validated by an experienced annotator. Using this manually validated annotation as the gold standard, we calculated the percentage of correct annotations in the initial partial annotation submitted to the annotator (cf. Table 4, columns LAS and UAS). Punctuation annotation was excluded, since punctuation marks are always attached to the root of the sentence. We also give the duration of manual annotation for each sample.

Sample	Model	Size	Coverage at	LAS	UAS	Duration of
		(tokens)	prob. level >0.7			man. annot.
viaule_1	it_isdt	352	84.7 %	81.2	88.7	30'
viaule_2	fr_partut+gsd+sequoia	325	86.5 %	74.8	85.2	32'
viaule_3	pt_bosque	337	88.3 %	84.5	89.4	21'

Table 4: Results of the manual annotation of new Occitan samples

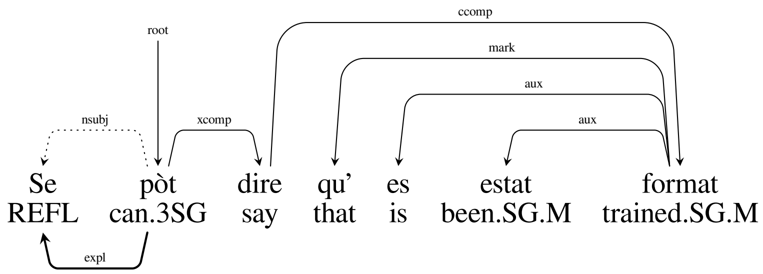
The elevated LAS and UAS scores show that the annotator's job consisted mostly in completing the partial annotation, whereas the actual corrections were less frequent, which is in line with our annotator's observations. The ergonomic value of such input is corroborated by the annotation duration times, which point towards a mean annotation speed of around 730 tokens/h. Since this annotator's speed during the annotation of the initial evaluation sample in Occitan was around 340 tokens/h, the utility of pre-processing with transferred models is certain. In order to verify if there were any noticeable differences between the outputs of different models, we proceeded to a more detailed analysis of the validation process.

5.2 Manual annotation analysis

Given the differences between the languages on which the three models were trained, we could expect some differences in their output. However, the three models performed in a largely consistent way: the annotator observed that in the three samples the internal structure of the NP was mostly well processed, whereas verbal dependents seemed to be more challenging.

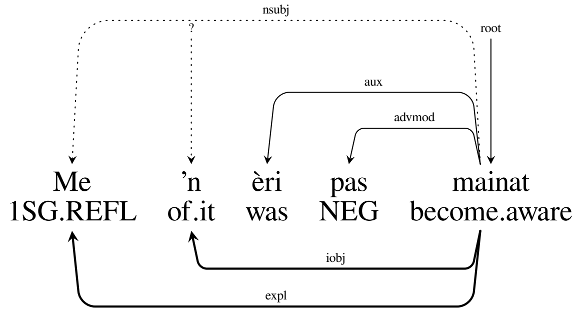
[5] Sèrgi Viaule: *Escorregudas en Albigés*. Lo Clusèl, 2012.

An issue related to lexical information occurred with reflexive pronouns: according to the UD guidelines, these should be treated as expletives with the *expl* syntactic label. However, given the minimal POS annotation in the Occitan corpus and the fact that the models had no access to lexical information, it was impossible to distinguish these pronouns from any others. They were therefore often annotated as nominal subjects, direct objects and indirect objects, which are common functions for other types of pronouns (cf. example 2)[6].



(2)
'You could say that he has been trained.'

In general, the annotation of pronouns proved difficult for the three models. Pronouns in sentence-initial position were often annotated as nominal subjects (*nsubj*), and in the case of pronominal clusters, pronouns other than the first often had no annotation, indicating that the dependencies produced by the parser were not sufficiently reliable to pass our filtering criteria (cf. example 3). This may not be surprising for the model trained on French, which has an obligatory subject, but it is for the ones learned on Italian and Portuguese, which allow the dropping of the subject.



(3)
'I hadn't noticed it.'

Although this type of error was recurrent, it was relatively easy to detect and correct.

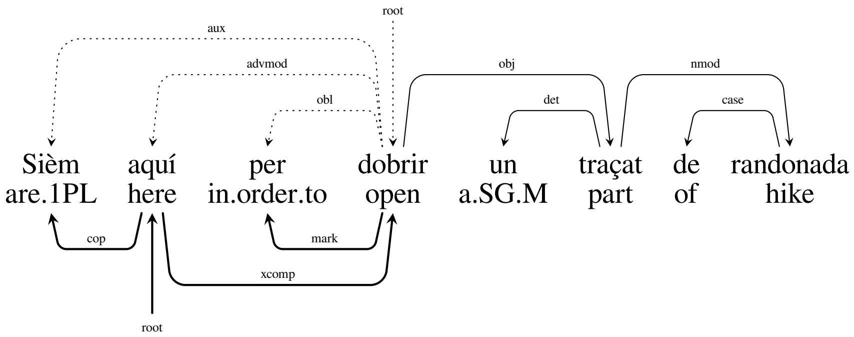
Another less frequent but interesting issue retained the attention of the annotator: the auxiliaries. The Occitan verb *èsser* 'to be' can behave both as a copula and as an auxiliary in complex verbal forms, and whereas both of these usages receive the tag AUX on the POS level, their treatment on the syntactic level differs. The auxiliaries receive the label *aux* and are governed by the main verb of the complex form. The copulas are typically treated as *cop* and governed by their complement, except for the cases where they introduce a clause (cf. *The problem is that this has never been tried*), in which case they are treated as the head of the structure and carry the label most appropriate to the context in which they appear[7].

The annotator noticed that the forms of *èsser* tended to be treated as auxiliaries even when they were in reality a copula, especially if there was a main verb in their proximity (cf. example 4).

Correcting these structures was particularly time consuming because the annotator not only had to correct the annotation of the verb *èsser*, but also to remove and then redo several other dependencies in its neighbourhood. This also applies to all cases of root miss-identification.

[6] In the following examples, the syntactic annotation produced by the model is given above the sentence, with the incorrect part marked by dotted arcs. The correct analysis of the incorrect arcs is given below the sentence, in boldface arcs. The dependencies missing from the original annotation are indicated as having no governor, with *?* as label.

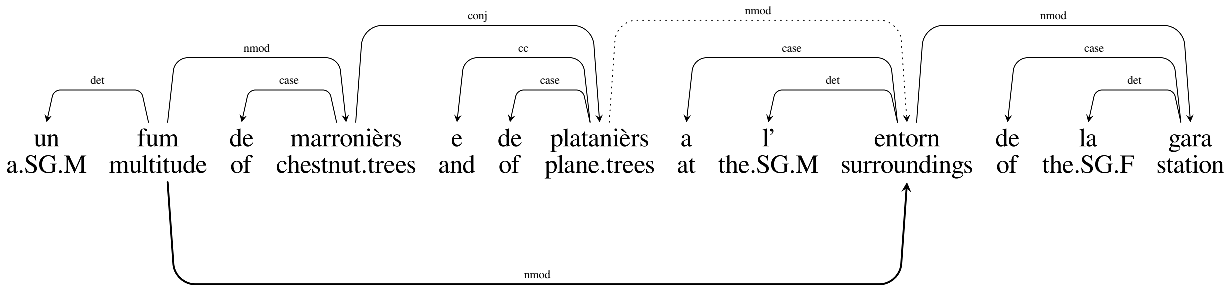
[7] Cf. the UD guidelines: `https://universaldependencies.org/u/dep/all.html#al-u-dep/cop`.


root
aux
advmod
obl
obj
det
nmod
case
Sièm aquí per dobrir un traçat de randonada
are.1PL here in.order.to open a.SG.M part of hike
cop
mark
xcomp
root

(4)

'We are here to open a part of a hike.'

More globally, all three models had difficulties with long-distance dependencies (cf. example 5)[8]. The models produced relatively few of them in each of the samples, and their accuracy rate was relatively low in two of the texts (cf. Table 5). However, it should be noted that this type of dependency is a long-standing issue in parsing and may not be due to model transfer.


conj
nmod
nmod
nmod
cc
case
case
det
case
case
det
det
case
det
un fum de marroniès e de plataniès a l' entorn de la gara
a.SG.M multitude of chestnut.trees and of plane.trees at the.SG.M surroundings of the.SG.F station
nmod

(5)

'a multitude of chestnut trees and plane trees around the station'

Sample	Model	Total long-distance deps.	Correct long-distance deps.
viaule_1	it_isd	18	12
viaule_2	fr_partut+gsd+sequoia	12	7
viaule_3	pt_bosque	13	10

Table 5: Long-distance dependency annotation per text sample

As mentioned above, some of these issues are undoubtedly related to the lack of lexical information in the models. Pronoun processing may be improved simply by including the pronoun type in the morphosyntactic traits of the corpus. This step is already planned for the next cycle of syntactic annotation. The issue with the distinction between the copulas and the auxiliaries is more complex, but even here, a presence of a morphosyntactic trait indicating the nature of the main verbs in the corpus (specifically, infinitive *vs* past participle) may contribute to the solution. This information will also be added to the corpus. Finally, the consistency of the output across the three models indicates that it could be useful to merge their training corpora and learn one global model, which is another direction we will be taking in the immediate future.

6 Conclusions and future work

In this paper we presented the application of cross-lingual dependency parsing on Occitan with the goal of accelerating the manual annotation of this language. 14 UD corpora of 8 Romance languages were used

[8] For the scope of this paper, we define long-distance dependencies as having 6 or more intervening tokens between the governor and the dependent.

to train 21 different delexicalized parsing models. These models were evaluated on a manually annotated Occitan sample. The top 5 models achieved LAS scores ranging from 70.0 to 71.6, and UAS scores from 75.3 to 76.0. They were trained on Italian, Portuguese and French. From the top 5 models, 3 were selected (one per language) and used to annotate new Occitan samples. These were then submitted to an experienced annotator for manual validation. The annotation speed in these conditions went from 340 tokens/h to 730 tokens/h and the annotator also reported greater facility in facing the task. Their observations show that the three models had largely consistent outputs, but they also note several recurring issues, such as erroneous processing of copula structures and pronouns, and problems in the identification of long-distance dependencies.

Some of these problems can be tackled by simple strategies. In order to improve pronoun and auxiliary processing, the morphosyntactic traits encoding the pronoun type and the nature of the verb form will be included in our corpora in the following annotation cycle. Given the consistent output of the three models, we will also combine their training corpora and learn one last global model in the hope of achieving further output improvements.

Regardless of these issues, the positive impact of the application of cross-lingual delexicalized parsing on the manual annotation of Occitan is clear. The annotation speed achieved by the annotator shows that they were able to almost double the amount of annotated text in around half the time needed to process the initial evaluation sample. Using the delexicalized models to pre-process the data also had an important ergonomic and psychological effect: the annotator noted that it was less daunting to correct the output of the models than to face completely blank sentences.

Finally, it is important to point out that this was a reasonably quick process. Since the goal was to accelerate manual annotation, this work had to be less costly than manual annotation itself. This condition was met: thanks to the general quality of the UD corpora and their documentation, the work described in this paper was an efficient exercise with satisfying results.

Acknowledgements

The work described in this paper is supported by the Linguatec project, a European project that is part of the Interreg POCTEFA 2017 Programme.

References

Delphine Bernhard, Anne-Laure Ligozat, Fanny Martin, Myriam Bras, Pierre Magistry, Marianne Vergez-Couret, Lucie Steiblé, Pascale Erhart, Nabil Hathout, Dominique Huck, et al. 2018. Corpora with part-of-speech annotations for three regional languages of france: Alsatian, Occitan and Picard. In *11th edition of the Language Resources and Evaluation Conference*.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 845–850.

Teresa Lynn, Jennifer Foster, Mark Dras, and Lamia Tounsi. 2014. Cross-lingual transfer parsing for low-resourced languages: An Irish case study. In *Proceedings of the First Celtic Language Technology Workshop*, pages 41–49.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the conference on empirical methods in natural language processing*, pages 62–72. Association for Computational Linguistics.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 92–97.

Avihai Mejer and Koby Crammer. 2012. Are you sure? confidence in prediction of dependency tree edges. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 573–576.

Aleksandra Miletic. 2018. *Un treebank pour le serbe: constitution et exploitations*. Ph.D. thesis, Université de Toulouse - Jean Jaurès.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Rudolf Rosa and David Mareček. 2018. Cuni x-ling: Parsing under-resourced languages in CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 187–196.

Olga Scrivner and Sandra Kübler. 2012. Building an Old Occitan corpus via cross-language transfer. In *KONVENS*, pages 392–400.

Jean Sibille. 2000. Ecrire l'occitan: essai de présentation et de synthèse. In *Les langues de France et leur codification. Ecrits divers–Ecrits ouverts,*. L'Harmattan.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. 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. Association for Computational Linguistics.

Jörg Tiedemann. 2014. Rediscovering annotation projection for cross-lingual parser induction. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1854–1864.

Jörg Tiedemann. 2015. Cross-lingual dependency parsing with universal dependencies and predicted PoS labels. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 340–349.

Assaf Urieli. 2013. *Robust French syntax analysis: reconciling statistical methods and linguistic knowledge in the Talismane toolkit*. Ph.D. thesis, Université Toulouse le Mirail-Toulouse II.

Marianne Vergez-Couret and Assaf Urieli. 2015. Analyse morphosyntaxique de l'occitan languedocien: l'amitié entre un petit languedocien et un gros catalan. In *TALARE 2015*.

Marianne Vergez-Couret. 2016. Description du lexique Loflòc. Technical report.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, et al. 2017. CoNLL 2017 shared task: multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21.

Developing Universal Dependencies for Wolof

Cheikh Bamba Dione
University of Bergen / Sydnesplassen 7, 5007 Bergen
dione.bamba1@uib.no

Abstract

This paper presents work on the creation of a Universal Dependency (UD) treebank for Wolof as the first UD treebank within the Northern Atlantic branch of the Niger-Congo languages. The paper reports on various issues related to word segmentation for tokenization and the mapping of PoS tags, morphological features and dependency relations to existing conventions for annotating Wolof. It also outlines some specific constructions as a starting point for discussing several more general UD annotation guidelines, in particular for noun class marking, deixis encoding, and focus marking.

1 Introduction

Wolof (ISO code: 693-3) is a Niger-Congo language mainly spoken in Senegal and Gambia.[1] Until recently, not many natural language processing (NLP) tools or resources were available for this language. Dione (2012a) developed a finite-state morphological analyzer. Dione (2014) reported on the creation of a deep computational grammar for Wolof based on the Lexical Functional Grammar (LFG) framework. That grammar has been used to create the first treebank for this language, making an important contribution to the development of the LFG parallel treebank (Sulger et al., 2013).

Treebanks play an increasingly important role in computational and arguably also theoretical linguistics. A treebank can be defined as a collection of sentences that typically contain various kinds of morphological and syntactic annotations (Abeillé, 2003). In recent years, different language processing applications (e.g. question answering, machine translation, information extraction) require high-quality parsers. Reliable and robust parsing models can be trained and induced from treebanks (Manning and Schütze, 1999).

The basic assumption in dependency grammar is that syntactic structure consists of lexical elements linked by binary asymmetrical relations called *dependencies* (Tesnière, 1959). The arguments to these relations consist of a head and a dependent. The head word of a constituent is the central organizing word of that constituent. The remaining words in the constituent are considered to be dependents of their head. Figure 1 shows an example of dependency structure from the WTB for the sentence[2] given in (1).

(1) *Noonu laa mujj a tànn beneen mecce, jàng dawal awiyoŋ.*
ADV 1SG.NSFOC finally.do to choose another profession learn pilot airplane
'So then I chose another profession, and learned to pilot airplanes.'

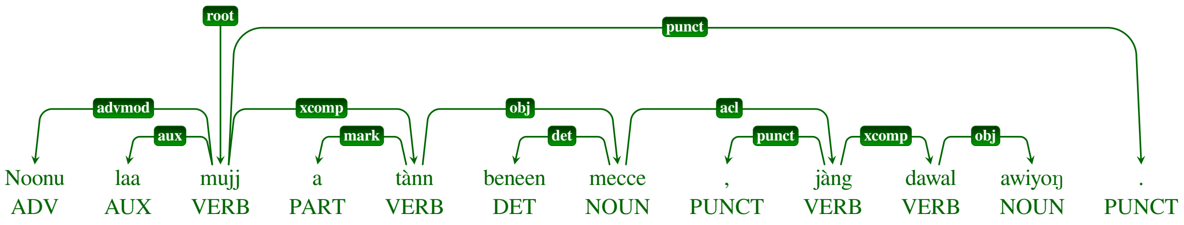


Figure 1: Example of a dependency structure from the WTB

[1] See http://www.ethnologue.com/language/WOL.

[2] Source: Wolof translations of *The Little prince* (Saint-Exupéry, 1971) available from http://www.wolof-online.com.

This paper presents work on the development of a Universal Dependency (UD) treebank for Wolof (henceforth WTB). It is the first effort in building dependency structures for Wolof in particular, and for the Northern Atlantic branch of the Niger-Congo languages in general. The annotations contained in the Wolof LFG treebank (henceforth WolGramBank) served as a basis for the creation of a scheme for the WTB. Note, however, that the WTB is not an automatic conversion from the LFG treebank, but was rather created manually (from scratch). This is mainly because such an automatic conversion (which is planed as future work) involves non-trivial mapping issues between LFG and UD. One of the most significant challenges is to determine which syntactic level of representation – constituency structure or functional structure – is the most natural basis for constructing dependency representations. Other crucial issues include e.g. the procedure of selecting the true head of syntactic constituents, the mapping from LFG to UD relations, the treatment of copula, coordination and punctuation (Meurer, 2017; Przepiórkowski and Patejuk, 2019).

The paper is structured as follows. Section 2 gives a brief overview of some salient features of the Wolof language. Section 3 describes the data collection process and the composition of the corpus. Section 4 discusses issues of word segmentation for tokenization. Section 5 describes the annotation processes for parts of speech (PoS), morphological features and syntactic relations. Section 6 concludes the discussion.

2 Background on Wolof

Before we take up the issue of the creation of a treebank for Wolof, we need to provide the reader with a general understanding of some salient features of that language.

2.1 Nouns, noun classes and determiners

Like the other Atlantic languages, Wolof has a noun class (NC) system (Greenberg, 1963; Sapir, 1971; McLaughlin, 1997) that consists of approximatively 13 noun classes:[3] 8 singular, 2 plural, 2 locative, and 1 manner noun classes. Like in Bantu languages, the Wolof noun class system also encodes *Number*. However, class membership is not marked on the noun itself, but rather on the noun dependents like determiners (e.g. articles, demonstratives), but also on (indefinite, interrogative and relative) pronouns and adverbs (locatives, manner). The noun classes are identified by their index (*b*, *g*, *j*, *k*, *l*, *m*, *s*, *w* for singular NCs and *y*, and *ñ* for plural NCs). The index "appears in the form of a single consonant on nominal dependents such as determiners and relative particles" (McLaughlin, 1997, p. 2).

Wolof determiners agree in noun class with the head noun. Determiners for different noun classes are distinguished by a consonant that is final (i.e. as a suffix) in the indefinite article (2c) and word-initial (i.e. as a prefix) in all other determiners. In addition, definite determiners encode information about proximity and distance with respect to the noun reference. As shown in (2), the definite article is constructed by suffixing a spatial deictic, *-i* for the proximal (2a) or *-a* for the distal (2b), to the consonantal class marker.[4]

(2) a. *xaj b-i*
dog NC-DFP
'the dog (proximal)'

b. *xaj b-a*
dog NC-DFD
'the dog (distal)'

c. *a-b xaj*
INDF-NC dog
'a dog'

Wolof has a rich system of demonstratives (Robert, 2016). These combine indications of the distance and reference point with respect to the speaker or addressee. For instance, for the *b* noun class, the four most commonly used forms are (*bii*, *bale*, *boobale*, and *boobu*), as exemplified in (3) with the noun *xaj* "dog".

(3) a. *xaj bii* 'this dog' (close to me, wherever you may be)
b. *xaj bale* 'that dog' (far away from me, wherever you may be)
c. *xaj boobale* 'that dog' (far away from both of us, but closer to you than to me)
d. *xaj boobu* 'that dog' (close to you and far away from me)

[3] The number of noun classes may vary according to dialects (Tamba et al., 2012).

[4] Abbreviations in the glosses: ADV: adverb; COP: copula; DEM: demonstrative; DET: determiner; DFP: definite proximal; DFD: definite distal; GEN: genitive; INDF: indefinite; LOC: locative; IPFV: imperfective; NC: noun class; NSFOC: non-subject focus; OBJ: object; POSS: possessive; PRES: present; PROG: progressive; PST: past tense; PL: plural; SG: singular; SFOC: subject focus; SUBJ: subject; VFOC: verb focus; 1, 2, 3: first, second, third person.

In Wolof, noun class membership is determined by a number of factors, including phonological, semantic and morphological criteria (McLaughlin, 1997; Tamba et al., 2012). For instance, many nouns that begin with [*w*] are in the *w*-class. Concerning morphology, nouns derived with certain derivational suffixes (e.g. *-in*) are assigned a specific class (e.g. the *w*-class). Finally, regarding semantics, trees typically are in the *g*-class, while most fruits are in the *b*-class. Also, the singular human noun class is the *k*-class, while the default plural human noun class is the *ñ*-class. However, the aforementioned factors just point to few tendencies found in the language. In fact, for each class, there are several words that do not follow these factors. The Wolof noun class system lacks semantic coherence (McLaughlin, 1997). The same can be said for the phonological and the morphological criteria. None of these factors are systematic indicators of noun classes in Wolof.

Furthermore, Wolof nouns are typically not inflected except for the genitive and the possessive case. Wolof genitives (4) are head-initial and show affinities with the Semitic construct state (Kihm, 2000). Such constructions involve a possessed entity described as the head and a possessor as its complement. The genitive relationship is overtly marked on the head noun by means of the *-u* suffix (e.g. *kër-u*) which precedes its complement (*buur* "king"). This suffix may also appear in other constructions like (5), which, unlike (4), do not denote possession, but rather seems to be just a normal compound, despite the similarity between these two constructions. In many other compounds like (6), the genitive marker does not appear at all.

(4) *kër-u buur*
house-GEN king
'king's house'

(5) *ndox-u taw*
water-GEN rain
'rain water'

(6) *téere xam-xam*
book knowledge
'knowledge book'

2.2 Adjectives

Wolof has no category for adjectives (Church, 1981; McLaughlin, 2004). The 'adjectival' concepts in Indo-European languages are typically expressed by stative verbs in Wolof. Adjectival constructions are realized as relative clause structures with the "adjective" being inflected like verbs.

2.3 Verbal system

In Wolof, a verb constituent has two components (Robert, 1991; Robert, 2000). The first component is the verb which is typically an invariant (unless derived) lexical stem. The second component is an inflectional marker that conveys the grammatical specifications of the verb, including person, number, tense, aspect, and mood features as well as the information structure of the sentence (focus). The inflectional marker can be preposed, postposed, or suffixed to the lexical stem, resulting in ten different paradigms or conjugations (Robert, 2010). Among these paradigms, we can distinguish non-focused conjugations from focused ones. Non-focus conjugations include perfective (7-8) and imperfective (9) constructions.

(7) *Xaj b-i lekk na.*
dog NC-DFP eat 3SG
'The dog has eaten.'

(8) *Lekk na.*
eat 3SG
'She/he/it has eaten.'

(9) *Xaj b-i di-na lekk.*
dog NC-DFP IPFV-3SG eat
'The dog will eat.'

Like Arabic (Attia, 2007) and many other languages, Wolof is a pro-language. This means that the subject can be explicitly stated as an NP or implicitly understood as a pro-drop. The pro-drop nature of the language is illustrated in the affirmative perfective examples given in (7-8). While (7) has an explicit subject, (8) does not. Nevertheless both sentences are grammatical. In (8), there is no overt subject, because the language freely allows the omission of such an argument. In examples (7-8), *na* is an agreement marker. It carries information about number, and person, which enables the reconstruction of the missing subject in (8).

Wolof has three focus conjugations: subject focus, verb focus, and complement focus. As these names imply, these constructions vary according to the syntactic function of the focused constituent: subject, verb, or complement. The latter has a wide meaning and refers in general to any constituent which is neither subject nor main verb. Table 1 illustrates the inflections for the verb *lekk* 'to eat' and the object *jën* 'fish' in the three focus types. As can be seen, focus is marked morphosyntactically.

The examples (10), (11) and (12) illustrate subject, verb and non-subject focus constructions, respectively.

	Subject focus	**Verb focus**	**Complement focus**
1SG	*maa* lekk jën	*dama* lekk jën	jën *laa* lekk
2	*yaa* lekk jën	*danga* lekk jën	jën *nga* lekk
3	*moo* lekk jën	*dafa* lekk jën	jën *la* lekk
1PL	*noo* lekk jën	*danu* lekk jën	jën *lanu* lekk
2	*yeena* lekk jën	*dangeen* lekk jën	jën *ngeen* lekk
3	*ñoo* lekk jën	*dañu* lekk jën	jën *lañu* lekk

Table 1: Subject, verb and complement focus in Wolof.

(10) *Faatu moo lekk jën.*
Faatu 3SG.SFOC eat fish
'It's Faatu who ate fish.'

(11) *Faatu dafa lekk jën.*
Faatu 3SG.VFOC eat fish
'What Faatu did is eat fish.'

(12) *Jën la Faatu lekk.*
fish 3SG.NSFOC Faatu eat
'It's fish that Faatu ate.'

Morphologically, one can reconstruct the origins of the subject, verb and non-subject focus markers as *-a*, *da-* and *la-*, respectively. An evidence for such a reconstruction can be seen in examples where the focus marker amalgamates with a noun or a proper name, as shown in (13a). Here, the form *Faatoo* is a phonological contraction and can be decomposed in *Faatu* + *a*, as illustrated in (13b). The main difference between (10) and (13a) is that in the former the constituent Faatu is dislocated, while in the latter that constituent bears the subject function. Indeed, (10) could be translated as "Faatu, it's her who ate the fish".

(13) a. *Faatoo lekk jën.*
Faatu.SFOC eat fish
'It's Faatu who ate fish.'

b. *Faatu a lekk jën.*
Faatu SFOC eat fish
'It's Faatu who ate fish.'

3 Data collection

The basis for the development of the WTB is a corpus of natural text data selected from the following sources: OSAD,[5] Wolof Online,[6] Wolof Wikipedia,[7] and Xibaaryi.com.[8] Table 2 lists the sources of the corpora used for creating the Wolof UD treebank.

Source	**Genres**	**# Docs**	**# Tokens**	**# Sentences**
OSAD	didactic, expository	6	6269	265
Wolof Online	informative, narrative	18	12988	673
Wolof Wikipedia	encyclopedic	12	9232	500
Xibaaryi	informative	17	15095	669

Table 2: Texts and genres in WTB.

The selection of texts for the WTB was meant to satisfy the following criteria. First, the data should be freely available as far as possible. Second, the text types should be chosen which are interesting to typical UD users. The data selected from Wikipedia is freely available under a Creative Commons license, facilitating its annotation and distribution. Also, users interested in computational linguistics, corpus linguistics and language typology may prefer texts which resemble other treebank texts or are even available in other languages, such as Wikipedia. Third, a range of different genres should be covered. Accordingly, we include texts from other sources than Wikipedia. For those sources, it was necessary to first clarify copyright issues.

4 Tokenization and word segmentation

Syntactic analysis in UD is based on a lexicalist view of syntax (i.e. dependency relations hold between words). According to De Marneffe (2014), practical computational models gain from this approach. Following this, the basic units of annotation are syntactic (not phonological or orthographic) words. Therefore, clitics attached to orthographic words need to be systematically segmented for proper syntactic analysis.

[5] http://www.osad-sn.com

[6] http://www.wolof-online.com

[7] https://wo.wikipedia.org

[8] http://www.xibaaryi.com

Word segmentation for tokenization in Wolof is a non-trivial task due to an extensive use of cliticization (Dione, 2017). As in Arabic (Attia, 2007), function words such as prepositions, conjunctions, auxiliaries and determiners can attach to other function or content words. Like Amharic (Seyoum et al., 2018), clitics in Wolof may undergo phonological changes. They may assimilate with word stems and with each other, making it difficult to recognize and handle them properly. The phonological change is also exhibited in the written form where clitics are attached to their host. For proper segmentation, then, we need to recover the underlying form first. For example, the word *cib* 'in a', can be segmented into the preposition *ci* 'in' and the indefinite article *ab* 'a'. However, if we simply segment the first characters *ci*, the remaining form, *b* will not have meaning. Furthermore, a non-trivial issue is ambiguity of clitics. For instance, a form like *beek* can be split into *bi* 'the' and *ak* where *ak* can actuallly be interpreted as a conjunction 'and' or a preposition 'with'.

Table 3 provides examples of full form words consisting of stems with clitics. The first row of the table is to be read as follows: the preposition *ak* 'with' may encliticize to the verbal stem *daje* 'meet', yielding the surface form *dajeek*.[9] The other surface forms involve different grammatical categories (determiners, conjunctions, pronouns, auxiliaries, etc.) and occur in a similar manner.

Stem PoS	***Clitic PoS***	***Example***	***Word form***	***Literal translation***
VERB	PREP	*daje* 'meet' + *ak* 'with'	*dajeek*	'meet with'
	DET	*joxe* 'give' + ay 'some'	*joxeey*	'give some'
DET	PREP	*ba* 'the' + *ak* 'with'	*baak*	'the with'
	CONJ	*bi* 'the' + *ak* 'and'	*beek*	'the and'
PREP	DET	*ci* 'in' + *ab* 'a'	*cib*	'in a'
	PREP	*ca* 'about' + *ak* 'with'	*caak*	'about with'
NOUN	CONJ	*ndox* 'water' + *ak* 'and'	*ndoxak*	'water and'
NAME	CONJ	*Ali* 'Ali' + *ak* 'and'	*Aleek*	'Ali and ...'
ADV	PRON	*fu* 'where' + *nga* 'you'	*foo*	'where you ...'
PRON	AUX	*ko* 'him/her' + *di*	*koy*	'him/her' + IPFV
	AUX	*mu* 3SG + *a* SFOC + *di* IPFV	*mooy*	3SG SFOC + IPFV
CONJ	AUX	*te* 'and' + *di* IPFV	*tey*	'and' + IPFV
	DET	*mbaa* 'or' + *ay* 'some'	*mbaay*	'or some'

Table 3: Examples of cliticization in Wolof

A crucial segmentation issue concerns the focus markers discussed in section 2.3. In accordance with the UD guidelines, we split the focus markers into a pronoun and a focus morpheme. Thus, contracted forms like third singular subject focus marker *moo* were decomposed into *mu* (3SG) and *a* (subject focus marker). The same applies for *dafa* which becomes *da* (verb focus marker) + *fa* (3SG), though *fa* is an irregular form. In contrast, *la* does not combine with a pronoun. The direct consequence of splitting focus elements like *moo* is that, as shown in (14b), the proper noun *Faatu* occurs in a dislocated position before the clause, and is resumed within the clause by the co-referential pronoun *mu*, the subject of the verb *lekk* 'eat'.

(14) a. *Faatu moo lekk jën.*
Faatu 3SG.SFOC eat fish
'It's Faatu who ate fish.'

b. *Faatu mu a lekk jën.*
Faatu 3SG SFOC eat fish
'Faatu, it's her who ate fish.'

Tokenization and word segmentation were done semi-automatically using the Wolof finite-state tokenizer (Dione, 2017). This tool includes a clitic transducer that can detect and demarcate contracted morphemes, handling these as separate words. For some cases, a manual revision was necessary.

5 Annotation

There are a number of existing interfaces in use that allow for manual annotation of UD treebanks. These include BRAT (Stenetorp et al., 2012), Arborator (Gerdes, 2013) and Tred.[10] In this work, manual annotation was done using UD Annotatrix (Tyers et al., 2018). Unlike the aforementioned tools, UD Annotatrix is designed specifically for Universal Dependencies. It can be used in online and in fully-offline mode. The tool is freely-available under the GNU GPL licence.

[9] The long vowel [*ee*] in *dajeek* results from a coalescence of the final vowel of *daje* with the stem-initial vowel of the PREP *ak*.

[10] https://ufal.mff.cuni.cz/tred/

5.1 Parts of speech annotation

The PoS tag set used in the UD scheme is based on the Universal PoS tag set (Petrov et al., 2012) and contains 17 tags. Because we wanted to use existing PoS tag annotation for Wolof as starting point, a mapping between the tagset in the Wolof LFG and the UD PoS tagset was necessary. At the coarse-grained level, the Wolof LFG tag set contains 24 tags. Thus, the conversion of the parts of speech information in LFG treebank to the UD PoS tag set required some considerations. Since UD does not allow sub-typing of PoS tags or language-specific tags, we adhere to this restriction. Below we discuss issues in adapting the UD annotation scheme to the existing Wolof tagset.

5.1.1 Nouns

WolGramBank makes a distinction between proper nouns and other noun types. One main reason for this is that proper nouns generally do not appear with determiners (while common nouns and indefinite pronouns for instance do). This distinction starts at early preprocessing steps (during tokenization and morphological analysis). The functional information about the syntactic type as a proper noun and the semantic type as a name are respectively provided by the morphological tags *+PropNoun* and *+PropTypeName*. Proper nouns are assigned the *NAME* tag, making the mapping to the corresponding UD tag *PROPN* straightforward.

Concerning the other noun types, WolGramBank distinguishes three categories: *NOUN*, *NGEN* and *NPOSS*. The first category includes nouns without any inflection (e.g. *këր* “house”). The second and third categories refer to nouns inflected in the genitive (e.g. *kër-u* “house of”) or in the possessive case (e.g. *kër-am* “his/her house”), respectively (see section 2.1).

In the WTB, all the three categories (common nouns, nouns inflected for genitive and those inflected for possessive) are mapped into the PoS category *Noun*. In terms of syntactic annotation, nouns with an apparent genitive marker are assigned the *nmod*[11] relation and are treated differently from those which do not show such an inflection, e.g. *téere* ‘book’ in (6). Nouns in the latter category are marked as *compound*. Using the UD features (FEATS), it was possible to further categorize the different forms, e.g. *Case=Gen* for the genitive and *Poss=Yes* for the possessive.

5.1.2 Determiners

In the WTB, determiners and quantifiers are assigned the *DET* category. A distinction between these categories can be made using features, e.g. *NumType=Card* for quantifiers, as it is done in some UD treebanks.

5.1.3 Adverbs

WolGramBank distinguishes between various types of adverbs, depending on whether an adverb modifies a verb, a clause, or introduces negation (e.g. negative particles). In the WTB, however, we define ADV for any kind of adverbs, and use the *Polarity* and *PronType* features (e.g. for relative/interrogative adverbs) to describe the type of adverb where necessary (the morphological features are discussed in section 5.2).

5.1.4 Verbs and auxiliaries

As discussed in section 2.3, Wolof verbs typically do not themselves carry inflectional markers. Instead, inflection is in many cases carried by so called inflectional elements that appear as separate words. The inflectional markers express a bunch of subject-related and clause-related features, including subject agreement, but also tense-aspect mood (TAM), polarity, and the focus in the sentence.

In the Wolof LFG Grammar, the inflectional markers are grouped under the category *INFL*. This category subdivides into four subcategories corresponding to the information whether the marker expresses subject focus, non-subject focus, verb focus and progressive. The *AUX* (for auxiliaries) tag is used mainly for the *di* imperfective marker (including its past tense inflected forms, e.g. *doon*). Furthermore, the tag *COP* is used for copula verbs and inflectional markers found in predicative constructions. This choice was motivated by the idea to provide a uniform analysis for both simple copula and clefts in Wolof, as both instantiate the same forms (Dione, 2012b).[12]

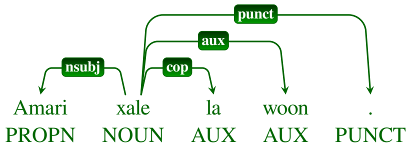
[11] *nmod* is used for nominal dependents of another noun and functionally corresponds to an attribute, or genitive complement.

[12] A more detailed discussion of the parallel syntactic proposed for copular and cleft clauses can be found in Dione (2012b).

However, the UD tagset scheme contains no *INFL* or *COP* tag. Still, it provides a general definition that allows for grouping these tags under the *AUX* category. UD defines an auxiliary as a function word that expresses grammatical distinctions not carried by the lexical verb, such as person, number, tense, mood, aspect. This is also the category provided for nonverbal *TAME* markers found in many languages. Thus, this is the category that fits the *INFL* tag from the Wolof LFG grammar. However, to keep the relevant information regarding the encoded information structure and copulae, it was necessary to introduce a new feature called *FocusType*. Such a feature is used to distinguish auxiliaries marking focus from other auxiliaries.

The UD guidelines state that the AUX category also includes copulas (in the narrow sense of pure linking words for nonverbal predication). Following this, the *COP* category from the LFG treebank was mapped to *AUX* in the Wolof UD treebank. This mapping, however, raised a small issue: in the UD scheme *AUX* cannot have a dependent, while in the existing annotation scheme for Wolof it is sometimes necessary for *COP* to have a dependent. An example is illustrated in (15) where the past tense particle *woon* has to be a dependent of the copula *la*. Following the UD practices, both the copular verb (e.g. *la*) and the tense particle (e.g. *woon*) have to be attached as siblings to the nonverbal predicate, as shown below.

(15) *Amari xale la woon.*
Amari child COP.3SG PAST
'Amari was a child.'



5.1.5 PRON

In WolGramBank, object and locative clitics (OLCs) are tagged as *CL* for clitics (Dione, 2013). A particular motivation for this was to distinguish these elements from subject pronouns, which are tagged as *PRON*. While subject pronouns have a predictable position in the sentence, OLCs have a quite special distribution, i.e. are special clitics according to Zwicky's definition (Zwicky, 1977).[13] First, they have a phrase structure position which is distinct from that of their non-clitic counterparts. While the latter typically follow the verb, the former usually precede it. Furthermore, OLCs have a set order amongst themselves. That is, if there is more than one clitic, they form a cluster. Considering these properties, OLCs are tagged as *CL* in WolGramBank. However, for UD compatibility reasons, both subject pronouns and object clitics are assigned the category *PRON* for pronouns. The relevant distinction is then made by using features, i.e. *Case=Nom* for subject clitics, and *Case=Acc* for object clitics. In contrast, locative clitics are assigned the *ADV* tag. Example (16) shows an instance of subject (*mu*), object (*ko*), and locative (*fa*) clitics.

(16) *Mu lekk ko fa.*
3SG.SUBJ eat 3SG.OBJ LOC
'So she/he/it eats it there.'

In addition, possessive, reflexive, relative, interrogative, demonstrative, and indefinite pronouns are also grouped under the *PRON* class. Like personal pronouns, possessive and reflexive pronouns have person and number features. Pronouns also include information about the noun class (where appropriate).

5.1.6 Adpositions

Wolof has only prepositions (no postpositions or circumpositions). The WolGramBank distinguishes between simple, partitive, and possessive prepositions. However, the UD convention does not further categorize prepositions, nor does it make a distinction between prepositions and postpositions. It rather recommends the category adposition (ADP) which is the cover term for both categories. Accordingly, in the WTB we use *ADP* without any subtype and that category actually only includes prepositions.

Table 4 shows the mapping between UD vs. WolGramBank PoS tags. It is a many-to-one (i.e. multiple WolGramBank tags mapping to one UD tag) rather than a many-to-many mapping, thus validating both annotation schemes. The WTB does not use the category *ADJ*, as the language has no adjectives.

[13] For an extensive discussion of Wolof object and locative clitics, see Zribi-Hertz and Diagne (2002).

UD PoS	Wolof Tagset	Example
ADP	PREP	*ci* 'in'
ADV	ADV	*léegi* 'now'
	CL	*fa* 'there'
AUX	AUX	*dina* 'I will'
	CL	*woon* (past tense particle)
	INFL	*a* (subj. focus marker)
CCONJ	CONJ	*ak* 'and' (nominal conjunction)
	CONJADV	*te* 'and' (clausal conjunction)
DET	DET	*bi* 'the'
	QUANT	*bépp* 'every'
INTJ	INTJ	*waaw* 'yes'
NOUN	NOUN	*kër* 'house'
	NPOSS	*këram* 'his house'
	NGEN	*këru* 'house of'
NUM	NUMBER	*fukk* 'ten'
PART	PART	*a* (infinitive particle)
PRON	PRON	*mu* (3SG subj. pron.)
	CL	*ko* (3SG obj. pron.)
PROPN	NAME	*Amari* 'Amari'
PUNCT	PUNCT	'.' period/full stop
SCONJ	COMP	*bu* 'when'
SYM	SYM	= (equal symbol)
VERB	VERB	*lekk* 'eat'
	COP	*di* 'to be'

Table 4: Mapping between the Wolof LFG and the UD PoS tagset

5.2 Morphological annotation

The UD annotation scheme defines a set of 23 morphological features across languages. These are divided into lexical vs. inflectional features. Lexical features such as *PronType* (pronoun type) and *Poss* (possessive) are attributes of lexemes or lemmas. Inflectional features are mostly features of individual word forms and are further subdivided into nominal features (e.g. *Gender*, *Case*, *Definite*) vs. verbal features (e.g. *Person*, *Number*, *Tense* and *Mood*). In contrast to the universal PoS tagset, the language specification allows treebanks to extend this set of universal features and add language-specific features when necessary.

One feature that is currently missing in the universal list of features and quite relevant for Wolof is *FocusType*. To capture the main distinction between the different focus constructions, we introduce *FocusType* as a new feature. This attribute can take three values: *subj*, *verb*, *compl* depending on the syntactic function of the constituent in focus. Another feature that needed to be updated was *NounClass*.[14] Although that feature is described in the UD guidelines, it was not used in any UD treebank so far, since UD currently does not contain any Bantu language. The description of *NounClass* indicates that the set of values of that feature is specific for a language family or group. The idea is to identify, within a language group, classes that have similar meaning across languages. However, one has to decide where the boundary of the group is.

The UD guidelines illustrate the use of the *NounClass* feature based on the system found in the Bantu language group. Following this, the feature has values that range from 1 to 20 noun classes called *Bantu1* to *Bantu20*. The class numbering system is accepted by scholars of the various Bantu languages and UD recommends the creation of similar numbering systems for the other families that have noun classes.

Because Wolof is not a Bantu language, and the Bantu classes were not extensible to Wolof, it was necessary to create a different set of classes (that could eventually be shared with some other related non-Bantu Niger-Congo languages). However, as mentioned above, one main difficulty with such an endeavour is the lack of semantic coherence in the Wolof noun class system. In most cases, and unlike in Bantu languages, there is no clear semantics, phonology or morphology that can explain the classification in Wolof.

The approach we adopted to tackle these issues was to create a set of classes for Wolof that follows a schema similar to the one proposed for Bantu languages. This means that the values of the feature had to be in a certain range (e.g. Wol1 - Wol13). It was also necessary to order the values in a way that would be comparable to the Bantu classes where possible.

[14] The NounClass feature is described in UD, since it is described in UniMorph (Sylak-Glassman, 2016).

To illustrate the numbering system in the Bantu languages, the UD guidelines listed 18 noun classes for Swahili. Some of these show a similarity with the Wolof noun classes, as illustrated in Table 5. For instance, the classes number 1 and 2 refer to singular and plural persons, respectively. It is easy to see that the Wolof equivalents of these two classes are the *k* and *ñ* class, respectively. Likewise, the classes number 7 and 8 have the typical meaning of singular and plural things, respectively. Their Wolof counterparts would be *l* and *y*, respectively. Thus, for these classes, it was not problematic to propose a comparable numbering system.

	Swahili	**Wolof**	
Class number	**Prefix**	**Affix**	**Typical Meaning**
1	*m-, mw-, mu-*	*k*	singular: persons
2	*wa-, w-*	*ñ*	plural: persons (a plural counterpart of class 1)
7	*ki-, ch-*	*l*	singular: things
8	*vi-, vy-*	*y*	plural: things (a plural counterpart of class 7)

Table 5: Noun system numbering for compatible classes between Bantu and Wolof.

However, for the remaining Wolof classes, a numbering system different from those found in Bantu was necessary. This is because the typical meaning of these Wolof classes did not match the semantics conveyed by the Bantu classes. Table 6 gives the numbering system proposed for Wolof (and eventually non-Bantu Niger-Congo languages). Also, as stated above, it is crucial to mention that the examples of typical meaning provided in this table are not meant to be reliable or systematic indicators of noun classes in Wolof. For each class, there are several words that do not follow these patterns. Also note that currently nouns are not marked with the *NounClass* feature. This is particularly motivated by the fact that nouns in Wolof (i) lack a class marker on the noun itself and (ii) may belong to several classes.

Class number	**Affix**	**Typical meaning**	**Value name**
1	*k*	singular: persons	Wol1
2	*ñ*	plural: persons	Wol2
3	*g*	singular: plants, trees	Wol3
4	*j*	singular: family members	Wol4
5	*b*	singular: fruits, default class	Wol5
6	*m*	singular: liquids	Wol6
7	*l*	singular: things	Wol7
8	*y*	plural: things	Wol8
9	*s*	singular: diminutive	Wol9
10	*w*	singular: no clear semantics	Wol10
11	*f*	locative	Wol11
12	*n*	manner	Wol12

Table 6: Noun class numbering for Wolof

As discussed in section 2.1, Wolof demonstratives encode information about deixis, including reference to the speaker and/or addressee. As with the *NounClass* feature, the *Deixis* feature is described in Unimorph (Sylak-Glassman, 2016), but not currently used by any UD treebank. So, to properly capture this information, the WTB introduced two features: *Deixis* and *DeixisRef*, which respectively represent deixis subdimensions corresponding to “Distance” and “Reference Point”. The distance distinction is a three-way contrast between proximate (*Prox*), medial (*Med*), and remote (*Remt*). Reference point is used to determine the relationship of the speaker, addressee, and referent of the pronoun. The latter dimension often overlaps with distance distinctions, but is sometimes explicitly separated. In the WTB, the two primary features for reference point are speaker as reference point (ref1), and addressee as reference point (ref2). Thus, the information contained in the Wolof demonstratives given in example (3) can be modeled as follows:

- close to me, wherever you may be ... Deixis=Prox|DeixisRef=1
- far from me, wherever you may be ... Deixis=Remt|DeixisRef=1
- far from both, closer to you ... Deixis=Med|DeixisRef=2
- close to you, far from me ... Deixis=Prox|DeixisRef=2

Table 7 summarizes the morphological features used in the WTB. PoS tags that do not have additional features, e.g. coordinating conjunctions (CCONJ), subordinating conjunctions (SCONJ), interjections (INTJ), particles (PART), proper names (PROPN), punctuations (PUNCT) and symbols (SYM), are not displayed.

UD PoS	Description	Morphological Features
ADP	Adpositions	Number=Sing,Plur; NounClass=Wol1,Wol2,..,Wol13;
ADV	Adverbs	Polarity=Neg,Pos; PronType=Rel,Int
AUX	Auxiliaries	Aspect=Hab,Imp,Perf,Prog; Focus=Subj,Verb,Compl; Mood=Cnd,Imp,Ind,Opt; Number=Sing,Plur; Person=0,1,2,3; Polarity=Neg,Pos; Tense=Fut,Past,Pres; VerbForm=Fin,Inf
NOUN	Nouns	Case=Gen; Poss=Yes
DET	Determiners	Definite=Def,Ind; Deixis=Prox, Med,Remt; DeixisRef=1,2; NounClass=Wol1,Wol2,..,Wol13; Number=Sing,Plur; Poss=Yes; PronType=Art,Dem,Int,Neg,Prs,Rel,Tot
NUM	Numerals	NumType=Card,Ord
PRON	Pronouns	Definite=Def,Ind; Deixis=Prox, Med,Remt; DeixisRef=1,2; NounClass=Wol1,Wol2,..,Wol13; Number=Sing,Plur; Poss=Yes; PronType=Art,Dem,Int,Neg,Prs,Rel,Tot
VERB	Non-auxiliary verbs	Aspect=Hab; Mood=Cnd,Imp,Ind; Number=Sing,Plur; Person=0,1,2,3; Polarity=Neg,Pos; Tense=Past,Pres; VerbForm=Fin,Inf

Table 7: Morphological features in the WTB

5.3 Syntactic annotation

The WTB uses most of the UD relations, apart from *amod*, *clf*, *dep*, *goeswith*, and *reparandum*. The two first relations are not relevant for Wolof, which lacks adjectival modifier[15] and classifier. Likewise, *goeswith* and *reparandum* are not used as the WTB data do not contain dysfluencies/orthographic errors. Finally, *dep* was irrelevant as it was always possible to determine a more precise relation. Table 8 lists the frequency of UD relations used in the WTB.

UD Relation	Description	Frequency	UD Relation	Description	Frequency
acl	clausal modifier of noun	123	expl	expletive	4
acl:relcl	relative clause modifier	2336	fixed	fixed MWEs	205
advcl	adverbial clause modifier	837	flat	flat MWEs	615
advmod	adverbial modifier	1446	iobj	indirect object	298
appos	appositional modifier	298	iobj:appl	indirect applied object	7
aux	auxiliary	3301	mark	marker	1835
case	case marking	2415	nmod	nominal modifier	1821
cc	coordinating conjunction	1367	nsubj	nominal subject	4395
ccomp	clausal complement	733	nummod	numeric modifier	377
compound	coumpound	220	obj	object	3318
compound:prt	phrasal verb particle	68	obj:appl	applied object	76
compound:svc	serial compound verb	75	obj:caus	causative object	118
conj	conjunction	1877	obl	oblique nominal	2138
cop	copula	626	obl:appl	applied oblique	79
csubj	clausal subject	50	orphan	orphan	13
det	determiner	3138	parataxis	parataxis	412
discourse	discourse elements	47	punct	punctuation	5319
dislocated	dislocated elements	548	xcomp	open clausal complement	928

Table 8: Universal dependency relations in WTB

6 Conclusion

This paper has presented the process of creating a Universal Dependency treebank for Wolof, the first UD treebank from the North Atlantic languages. Wolof is also the second Atlantic-Congo language (after Yoruba) that has a UD treebank. Adopting UD to existing conventions for annotating Wolof required several decisions to be made. We have discussed issues related to tokenization pointing out the challenge of clitic segmentation. We indicated that Wolof orthographic words may carry morphological information as well as other function elements of syntactic relations. The discussion has also shown that there are a number of challenges in adapting the UD scheme for Wolof. In particular we advocate the introduction of missing features for focus marking and deixis information, and the redefinition of the existing noun class feature for non-Bantu languages. In future, we plan to address the issue of automatic conversion of WolGramBank.

[15] The *amod* relation is only used to annotate foreign material (e.g. French texts) that is contained in the WTB.

Acknowledgements

I would like to thank the UD community, in particular Dan Zeman for many fruitful discussions. I also want to thank the anonymous reviewers for valuable comments and suggestions.

References

Anne Abeillé, editor. 2003. *Treebanks: Building and Using Parsed Corpora*. Kluwer.

Mohammed A Attia. 2007. Arabic Tokenization System. In *Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources*, pages 65–72. Association for Computational Linguistics.

Eric D. Church. 1981. *Le système verbal du wolof*. Faculté des Lettres et Sciences Humaines (FLSH), Université de Dakar.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In *LREC*, volume 14, pages 4585–4592.

Cheikh M. Bamba Dione. 2012a. A Morphological Analyzer For Wolof Using Finite-State Techniques. In *Proceedings of the Eigth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. ELRA.

Cheikh M. Bamba Dione. 2012b. An LFG Approach to Wolof Cleft Constructions. In Miriam Butt and Tracy Holloway King, editors, *The Proceedings of the LFG '12 Conference*, Stanford, CA. CSLI Publications.

Cheikh M. Bamba Dione. 2013. Handling Wolof Clitics in LFG. In Christine Meklenborg Salvesen and Hans Petter Helland, editors, *Challenging Clitics*, Amsterdam. John Benjamins Publishing Company.

Cheikh M Bamba Dione. 2014. LFG parse disambiguation for Wolof. *Journal of Language Modelling*, 2(1):105–165.

Cheikh M. Bamba Dione. 2017. Finite-state tokenization for a deep wolof lfg grammar. *Bergen Language and Linguistics Studies*, 8(1).

Kim Gerdes. 2013. Collaborative dependency annotation. In *Proceedings of the second international conference on dependency linguistics (DepLing 2013)*, pages 88–97.

Joseph H Greenberg. 1963. Some universals of grammar with particular reference to the order of meaningful elements. *Universals of language*, 2:73–113.

Alain Kihm. 2000. Wolof Genitive Constructions and the Construct State. In J. Lowenstamm & U. Shlonsky Lecarme, J., editor, *Research in Afro-Asiatic grammar: papers from the third conference on Afroasiatic languages*, pages 150–181. Amsterdam & Philadelphia : John Benjamins Publishing Co.

Christopher D Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Fiona McLaughlin. 1997. Noun classification in Wolof: When affixes are not renewed. *Studies in African Linguistics*, 26(1).

Fiona McLaughlin. 2004. Is there an adjective class in Wolof? In R.M.W. Dixon and Alexandra Y. Aikhenvald, editors, *Adjective classes. A crosslinguistic typology.*, pages 242–262. Oxford University Press.

Paul Meurer. 2017. From LFG structures to dependency relations. *Bergen Language and Linguistics Studies*, 8(1).

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Adam Przepiórkowski and Agnieszka Patejuk. 2019. From lexical functional grammar to enhanced universal dependencies. *Language Resources and Evaluation*, Feb.

Stéphane Robert. 1991. Approche énonciative du système verbal: le cas du wolof. *Editions du CNRS*.

Stéphane Robert. 2000. Le verbe wolof ou la grammaticalisation du focus. Louvain: Peeters, Coll. Afrique et Langage, 229-267. Version non corrigée.

Stéphane Robert. 2010. Clause chaining and conjugations in wolof. *Clause Linking and Clause Hierarchy: Syntax and Pragmatics*, 121:469–498.

Stéphane Robert. 2016. Content question words and noun class markers in wolof: reconstructing a puzzle. *Frankfurt African Studies Bulletin*, 23:123–146.

Antoine de Saint-Exupéry. 1971. Le petit prince. 1943. *Paris: Harvest.*

David J. Sapir. 1971. West Atlantic: an inventory of the languages, their noun class systems and consonant alternation. *Current Trends in Linguistics*, 7(1):43–112.

Binyam Ephrem Seyoum, Yusuke Miyao, and Baye Yimam Mekonnen. 2018. Universal dependencies for amharic. In *LREC*.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. 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. Association for Computational Linguistics.

Sebastian Sulger, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, Agnieszka Patejuk, Özlem Çetinoğlu, I Wayan Arka, and Meladel Mistica. 2013. ParGramBank: The ParGram Parallel Treebank. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 759–767, Sofia, Bulgaria.

John Sylak-Glassman. 2016. The composition and use of the universal morphological feature schema (unimorph schema). Technical report, Technical report, Department of Computer Science, Johns Hopkins University.

Khady Tamba, Harold Torrence, and Malte Zimmermann. 2012. Wolof quantifiers. In *Handbook of Quantifiers in Natural Language*, pages 891–939. Springer.

Lucien Tesnière. 1959. Eléments de syntaxe structurale. *Klincksieck, Paris.*

Francis M Tyers, Mariya Sheyanova, and Jonathan North Washington. 2018. UD Annotatrix: An annotation tool for universal dependencies. In *Proceedings of the 16th Conference on Treebanks and Linguistic Theories.*

Anne Zribi-Hertz and Lamine Diagne. 2002. Clitic placement after syntax: evidence from Wolof person and locative markers. *Natural Language & Linguistic Theory*, 20(4):823–884.

Arnold Zwicky. 1977. On Clitics. *Indiana University Linguistics Club.*

Improving UD processing via satellite resources for morphology

Kaja Dobrovoljc
Jožef Stefan Institute
Ljubljana, Slovenia
kaja.dobrovoljc@ijs.si

Tomaž Erjavec
Jožef Stefan Institute
Ljubljana, Slovenia
tomaz.erjavec@ijs.si

Nikola Ljubešić
Jožef Stefan Institute
Ljubljana, Slovenia
nikola.ljubesic@ijs.si

Abstract

This paper presents the conversion of the reference language resources for Croatian and Slovenian morphology processing to UD morphological specifications. We show that the newly available training corpora and inflectional dictionaries improve the baseline stanfordnlp performance obtained on officially released UD datasets for lemmatization, morphology prediction and dependency parsing, illustrating the potential value of such satellite UD resources for languages with rich morphology.

1 Introduction

Many treebanks and tools are nowadays available for natural language processing tasks based on the Universal Dependencies (UD) framework, aimed at cross-linguistically consistent treebank annotation to facilitate multilingual parser development, cross-lingual learning, and parsing research (Nivre et al., 2016). As shown by the two successive CoNLL shared tasks on multilingual parsing from raw text to UD (Zeman et al., 2017, 2018), existing UD systems achieve state-of-the-art results both in terms of dependency parsing and lower levels of grammatical annotation.

However, in addition to the officially released UD treebanks with complete syntactic and morphological annotations, the rapidly emerging UD tools would benefit from other language resources, as well. This is especially true for morphological annotation (lemmatization, PoS tagging and morphological feature prediction), as many languages employ much larger morphology-annotated corpora than the costly (sub)corpora annotated for syntax, as well as morphological lexicons, essential for high-quality processing of languages with complex morphology.

Examples of such cases are Croatian and Slovenian, two South Slavic languages with rich inflection. Their official UD releases include the conversions of the largest syntactically annotated corpora available for each language (Agić and Ljubešić, 2015; Dobrovoljc et al., 2017a), however, other manually created resources, such as the larger morphologically annotated corpora (Ljubešić et al., 2018b; Krek et al., 2019) and inflectional lexicons (Ljubešić, 2019; Dobrovoljc et al., 2019), have also been developed to support the development of related NLP tools (Ljubešić and Erjavec, 2016; Grčar et al., 2012) in the past.

The aim of this paper is to present the conversion of these resources to the UD formalism and explore their potential contribution to the state-of-the-art in UD processing for both languages, from lemmatization to morphology and syntax prediction. Using the stanfordnlp tool, we investigate the impact of newly available data on all three tasks by (1) retraining the tagging and lemmatization models on larger training sets and (2) performing a simple lexicon lookup intervention in the lemmatization procedure.

This paper is structured as follows. We first briefly describe the creation and the content of the newly released resources for both languages in Section 2, followed by the presentation of the experiments for their evaluation in Section 3. We present the corresponding results in Section 4 and conclude in Section 5 by a short discussion of their wider implications for related UD languages and the UD community in general.

2 Extending the resources for UD morphology

This section describes the development, the content and the availability of the extended UD resources for Slovenian and Croatian, namely the larger training sets for UD morphology (the ssj500k and hr500k corpora) and the large-scale UD-compliant lexicons of inflected forms (Sloleks and hrLex). Given the methodological differences in resource development for both languages due to divergent project frameworks and scopes, we present the resources by language rather than type. However, a brief quantitative overview and comparison is given at the end of the section.

2.1 Slovenian resources

Both the ssj500k training corpus (Erjavec et al., 2010) and the Sloleks lexicon of inflected forms (Dobrovoljc et al., 2017b) adopt the JOS morphosyntactic annotation scheme (Erjavec and Krek, 2008), compatible with MULTEXT-East morphosyntactic specifications (Erjavec, 2012), which define the part-of-speech categories for Slovene, their morphological features (attributes) and values, and their mapping to morphosyntactic descriptions (MSDs).[1] An automatic rule-based mapping from JOS to UD part-of-speech tags and features had already been developed as part of the original Slovenian UD Treebank conversion from the syntactically annotated subset of the ssj500k corpus (Dobrovoljc et al., 2017a), with the conversion scripts now publicly available at the CLARIN.SI GitHub repository.[2]

The large majority of conversion rules for morphology are direct mappings of specific categories – e.g. conversion of JOS numerals (`M`) with `Form=letter` and `Type=ordinal` to UD adjectives (`ADJ`) with feature `NumType=Ord` – making them directly applicable for converting any language resource with JOS morphosyntactic annotations, such as the resources presented in this paper. The only exception are the two rules involving predefined lists of JOS pronouns and adverbs to be converted to UD determiners (e.g. *ta* 'this' or *veliko* 'many'), which have been updated so as to cover the previously unknown vocabulary emerging from ssj500k and Sloleks (i.e. adding 135 new lemmas to the list of UD determiners).

2.1.1 ssj500k corpus

The ssj500k training corpus is the largest training corpus for Slovenian, with approx. 500,000 tokens manually annotated on the levels of tokenization, segmentation, lemmatization and morphosyntactic tagging. Variously-sized ssj500k subsets have also been annotated for other linguistic layers, namely named entities, JOS dependency syntax, semantic roles, verbal multi-word expressions and Universal Dependencies.

To extend UD morphology annotations to the entire ssj500k corpus, v2.1 of the corpus (Krek et al., 2018) was converted using the pipeline referenced above. Specifically, the script `tei2ud.xsl` takes the original XML TEI format as input, converts it to a CONLL-like tabular format with English JOS tags, features and dependencies, followed by the conversion to the standardized CONLL-U file with UD PoS and morphological features. This second step is performed by the `jos2ud.pl` script, which takes two mapping files as parameters, one for the PoS mapping (`jos2ud-pos.tbl`), and the other for feature mapping (`jos2ud-features.tbl`).

For occurrences of the verb *biti* ('to be') – the only instance of the PoS mapping depending on syntactic role – an additional set of scripts is applied (`add-biti-*.pl`) to disambiguate between the auxiliary, copula (both `AUX` in UD) and other (`VERB` in UD) usages of this verb, which are always labelled as an auxiliary verb in JOS. In contrast to the occurrences within syntactic trees enabling rule-based disambiguation and the unambiguous occurrences of *biti* preceding verbal participles (and potentially intervening pronous, adverbs, particles or conjunctions), the remaining 11,925 *biti* tokens in ssj500k have been disambiguated manually. This was performed by trained native speakers, with two annotators per decision and a third one in case of competing annotations (93.9% agreement, Cohen's Kappa 0.78).

The resulting ssj500k corpus with UD PoS tags, morphological features and their values has been released as part of ssj500k release v2.2 (Krek et al., 2019) under CC BY-NC-SA 4.0. In addition to the CONLL-U format, in which underscores have been inserted where the dependency annotations are missing, the information on UD morphology and syntax has also been added to the original TEI XML format with other types of annotation and metainformation, as illustrated in Figure 1.

The sentence element (`<s>`) contains words (`<w>`), punctuation symbols (`<pc>`) and whitespace (`<c>`), as well as segments (`<seg>`) for annotating spans of tokens, and link groups (`<linkGrp>`) for annotating

[1] The latest (Version 6) MULTEXT-East multilingual morphosyntactic specifications are available at `http://nl.ijs.si/ME/V6/` and being developed at `https://github.com/clarinsi/mte-msd`.

[2] `https://github.com/clarinsi/jos2ud`

```
<s xml:id="ssj1.1.2">
   <w ana="mte:Ncmsn" msd="UposTag=NOUN|Case=Nom|Gender=Masc|Number=Sing"
      lemma="dogodek" xml:id="ssj1.1.2.t1">Dogodek</w><c> </c>
   <w ana="mte:Sl" msd="UposTag=ADP|Case=Loc"
      lemma="v" xml:id="ssj1.1.2.t2">v</w><c> </c>
   <seg type="name" subtype="loc">
      <w ana="mte:Npmsl" msd="UposTag=PROPN|Case=Loc|Gender=Masc|Number=Sing"
      lemma="Ankaran" xml:id="ssj1.1.2.t3">Ankaranu</w>
   </seg><c> </c>
   <w ana="mte:Va-r3s-n"
      msd="UposTag=AUX|Mood=Ind|Number=Sing|Person=3|Polarity=Pos|Tense=Pres|
      VerbForm=Fin"
      lemma="biti" xml:id="ssj1.1.2.t4">je</w><c> </c>
   <w ana="mte:Va-p-sf" msd="UposTag=AUX|Gender=Fem|Number=Sing|VerbForm=Part"
      lemma="biti" xml:id="ssj1.1.2.t5">bila</w><c> </c>
   <w ana="mte:Agpfsn" msd="UposTag=ADJ|Case=Nom|Degree=Pos|Gender=Fem|Number=Sing"
      lemma="dramatičen" xml:id="ssj1.1.2.t6">dramatična</w><c> </c>
   <w ana="mte:Ncfsn" msd="UposTag=NOUN|Case=Nom|Gender=Fem|Number=Sing"
      lemma="nesreča" xml:id="ssj1.1.2.t7">nesreča</w>
   <pc ana="mte:Z" msd="UposTag=PUNCT" xml:id="ssj1.1.2.t8">.</pc>
   <linkGrp corresp="#ssj1.1.2" targFunc="head argument" type="UD-SYN">
     <link ana="ud-syn:root" target="#ssj1.1.2 #ssj1.1.2.t1"/>
     <link ana="ud-syn:case" target="#ssj1.1.2.t3 #ssj1.1.2.t2"/>
     <link ana="ud-syn:nmod" target="#ssj1.1.2.t1 #ssj1.1.2.t3"/>
     <link ana="ud-syn:aux" target="#ssj1.1.2.t1 #ssj1.1.2.t4"/>
     <link ana="ud-syn:cop" target="#ssj1.1.2.t1 #ssj1.1.2.t5"/>
     <link ana="ud-syn:amod" target="#ssj1.1.2.t7 #ssj1.1.2.t6"/>
     <link ana="ud-syn:nsubj" target="#ssj1.1.2.t1 #ssj1.1.2.t7"/>
     <link ana="ud-syn:punct" target="#ssj1.1.2.t1 #ssj1.1.2.t8"/>
   </linkGrp>
   <linkGrp corresp="#ssj1.1.2" targFunc="head argument" type="JOS-SYN">
      <link ana="jos-syn:Atr" target="#ssj1.1.2.t5 #ssj1.1.2.t1"/>
      <link ana="jos-syn:Atr" target="#ssj1.1.2.t3 #ssj1.1.2.t2"/>
      <link ana="jos-syn:Atr" target="#ssj1.1.2.t1 #ssj1.1.2.t3"/>
      <link ana="jos-syn:PPart" target="#ssj1.1.2.t5 #ssj1.1.2.t4"/>
      <link ana="jos-syn:Root" target="#ssj1.1.2 #ssj1.1.2.t5"/>
      <link ana="jos-syn:Atr" target="#ssj1.1.2.t7 #ssj1.1.2.t6"/>
      <link ana="jos-syn:Sb" target="#ssj1.1.2.t5 #ssj1.1.2.t7"/>
      <link ana="jos-syn:Root" target="#ssj1.1.2 #ssj1.1.2.t8"/>
   </linkGrp>
   <linkGrp corresp="#ssj1.1.2" targFunc="head argument" type="SRL">
      <link ana="srl:ACT" target="#ssj1.1.2.t5 #ssj1.1.2.t1"/>
      <link ana="srl:PAT" target="#ssj1.1.2.t5 #ssj1.1.2.t7"/>
   </linkGrp>
</s>
```

Figure 1: The full TEI encoding of the sentence *Dogodek v Ankaranu je bil dramatična nesreča.* ('The incident in Ankaran was a dramatic accident.').

links between tokens. The words contain annotation on their JOS (MULTEXT-East) morphosyntactic description (the `@ana` attribute), as well as the Universal Dependencies morphosyntactic features (`@msd`), the lemma of words (`@lemma`) and the ID of each token (`@xml:id`). The fact that a segment denotes a named entity is signaled by `@type="name"`, and the type of the named entity by the `@subtype` attribute. The Universal Dependencies syntactic relations are encoded in the `<linkGrp type="UD-SYN">` element, where the individual links give the head and argument of the relation, which is encoded in the `@ana` attribute. Note that the sentence identifier serves as a proxy for the virtual syntactic root of the sentence tree. Similarly, the JOS syntactic relations are encoded in the `<linkGrp type="JOS-SYN">` element. Finally, the semantic role relations are encoded in the `<linkGrp type="SRL">` element.

2.1.2 Sloleks morphological lexicon

The Sloleks morphological lexicon is the largest manually created collection of inflected forms in Slovenian, consisting of 2,792,003 inflected forms and 100,805 lemmas, with each inflected form bearing information on its lemma, grammatical features, pronunciation and frequency of usage. Version 1.2 of the lexicon (Dobrovoljc et al., 2015) has been converted using the same JOS-to-UD conversion script, which allows switching between corpus and lexicon mode. The converted lexicon with UD PoS tags (UPOS), features and values (FEATS) has been released as part of the Sloleks release 2.0 (Dobrovoljc et al., 2019) under CC BY-NC-SA 4.0, in the form of a tab-separated file listing the inflected form, its lemma, JOS MSD tag, frequency of usage, JOS PoS and features, and UD PoS and features. The mapping to the original Sloleks release in LMF XML encoding with several additional layers of information, such as pronunciation, is not explicit, but can be reproduced based on unique combinations of the given features.

2.2 Croatian resources

The hr500k training corpus (Ljubešić et al., 2018b) and the hrLex inflectional lexicon (Ljubešić, 2019) were developed on the margins of many projects, with the ReLDI project[3] giving the final push for their consolidation and publication.

The enrichment of these resources with UD information was format-wise very similar to that of the Slovenian resources described in Section 2.1, with (1) differences in the mapping of MULTEXT-East morphosyntactic annotations to the Universal Part-of-Speech (UPOS) and morphological features (FEATS) due to a slightly different tagset for Croatian and (2) no mappings performed on the dependency syntax level, as the corpus was manually annotated with the UD dependency syntax layer.

2.2.1 hr500k training corpus

The hr500k training corpus contains tokens manually annotated on the levels of tokenisation, sentence segmentation, morphosyntactic tagging, lemmatization and named entities. About half of the corpus is also manually annotated with UD syntactic dependencies. Furthermore, about a fifth of the corpus is annotated with semantic role labels. This corpus is considered to be the reference training corpus for Croatian. The details on the content of the corpus are described in Ljubešić et al. (2018a).

The morphosyntactic layer of the corpus was initially annotated with the MULTEXT-East morphosyntactic specifications (Erjavec, 2012) and the mapping to the UPOS and FEATS layers was performed semi-automatically, with the automatic part consisting of (1) applying an explicit mapping between MULTEXT-East tags and UPOS and FEATS tags[4] and (2) fallback to additional rules for pronouns and determiners, adverbs, numbers and the negated auxiliary.[5] The only non-automatic part of the mapping was the resolution of the category of abbreviations from MULTEXT-East to the corresponding parts-of-speech.

The resulting hr500k corpus was part of the initial release of hr500k (v1.0) and was published under CC BY-SA 4.0 (Ljubešić et al., 2018b).

2.2.2 hrLex inflectional lexicon

The hrLex inflectional lexicon (Ljubešić, 2019) is currently the largest inflectional lexicon of Croatian. The process of semi-automatically building the hrLex inflectional lexicon is described in Ljubešić et al. (2016).

[3] `https://reldi.spur.uzh.ch`

[4] `https://github.com/nljubesi/hr500k/blob/master/mte5-udv2.mapping`

[5] `https://github.com/vukbatanovic/SETimes.SR/blob/master/msd_mapper.py`

The mapping of the MULTEXT-East tags that were initially present in the lexicon to the UPOS and FEATS layers was performed by applying the mapping that was used to map the hr500k training corpus to these layers, without the need for the manual mapping.

The UD information became part of the hrLex lexicon with version 1.3 (Ljubešić, 2019), when the lexicon was published under the CC BY-SA 4.0 license. The lexicon is published as a tab-separated file listing the inflected form, its lemma, MULTEXT-East tag, MULTEXT-East morphological features, UPOS, FEATS, and the absolute and relative (per-million) frequency of usage in the hrWaC corpus (Ljubešić and Klubička, 2016).

2.3 Quantitative overview

This section gives a quantitative overview of the newly available resources for both languages, to illustrate their morphological complexity and the importance of the corresponding disambiguation in the process of morphological annotation and lemmatization (Section 3).

Table 1 shows, for the Slovene and Croatian corpora, first the number of tokens and types, where the latter is taken to be triplets consisting of the lower-cased wordform (i.e. token), lemma, and the MULTEXT-East XPOS (giving both PoS and features). This is followed by the numbers of each of the individual members of the triplet. As can be seen, both corpora have approximately half a million tokens, and somewhat under 100,000 lexical types, with the Croatian resource being somewhat smaller and having a poorer lexicon, most likely because of its more restricted variety of source texts. The Croatian corpus also uses almost half less tags, however, this follows from the overall smaller number of defined tags, as will be shown in the discussion of the lexicon.

Next are shown the numbers of out-of-vocabulary tokens and types against the two lexicons, Sloleks and hrLex, but not taking into account punctuation, which is not part of the lexicon. The Croatian corpus has almost twice as many OOV types and tokens, which is due to the construction of the Slovene lexicon, discussed below.

The last column gives the type ambiguity in the corpora, i.e. on the average, how many different interpretations (lemmas or tags) does each distinct wordform have. In both cases the number is very similar, 5/4. This means that, on average, each fourth word will have two interpretations, which is a simplified view of ambiguity, as some distinct wordforms have more than two interpretations.

	Tokens	**Types**	**Wforms**	**Lemmas**	**Tags**	**OOV types**	**OOV toks**	**Ambig.**
ssj500k	586,248	98,641	78,707	38,818	1,304	5.26%	18.33%	1.25
hr500k	506,457	84,789	66,797	34,321	768	9.70%	27.17%	1.27

Table 1: Size of newly available corpora for UD morphology.

Table 2 gives a quantitative overview of the two lexicons. The number of entries is the number of wordform / lemma / tag triplets, and the next three columns give, as with the corpora, the individual numbers of wordforms, lemmas and morphosyntactic tags. As can be seen, hrLex is almost twice as large as Sloleks, however, it does distinguish only about half the tags compared to Slovene. This is mostly due to the tags related to the dual number in Slovene and a very fine-grained typology of Slovene pronouns, which account for almost half of the tagset. This is also evidenced by the number of tags used on the Slovene corpus (1,304), which, although much larger than for Croatian, is much smaller than the lexicon inventory.

The last column gives the ambiguity in the lexicon, i.e. how many different interpretations in terms of lemma and tag does, on the average, one wordform have. As can be seen, this number is over three in both cases, with the Croatian ambiguity being significantly higher; we discuss the reasons below. It can also be noticed that the lexicon ambiguity is in both cases much greater than in the corpora, which is due to the fact that the lexicons contain the complete inflectional paradigms, although some of its word forms are rarely present in the corpora.

Figure 2 gives — on a logarithmic scale — the lemma sizes of the two lexicons by the UD part-of-speech. The most striking features are the significantly greater number of adverbs, adjectives and proper nouns of the Croatian lexicon. This stems from the automatic inclusion of adverbs derived from adjectives,

	Entries	Wforms	Lemmas	Tags	Ambig.
Sloleks	2,792,003	921,869	96,593	1,900	3.03
hrLex	6,427,709	1,697,943	164,206	900	3.79

Table 2: Size of newly available lexica for UD morphology.

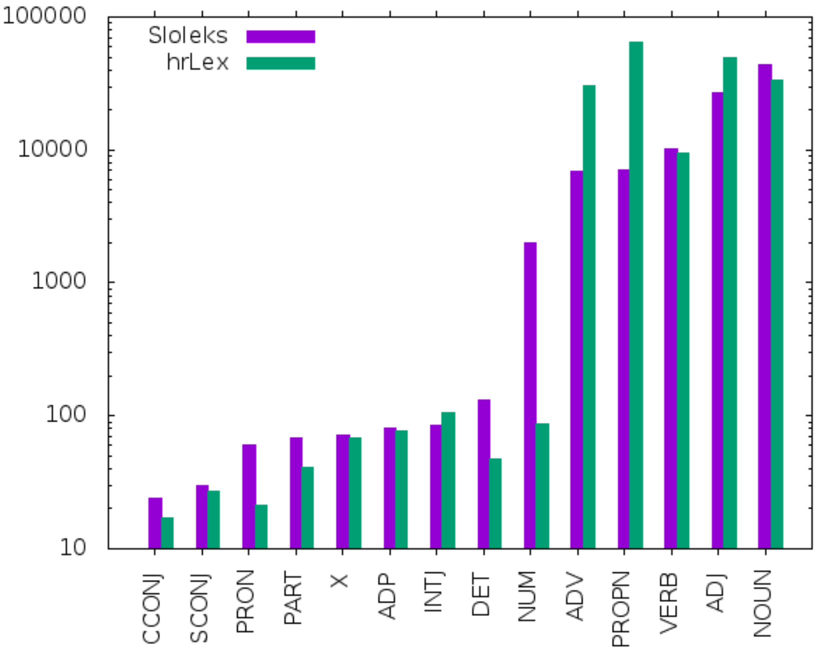


Figure 2: Size of lexicons by UD part-of-speech

and possessive adjectives derived from nouns in hrLex, and, of course, the preference given to including a large number of proper nouns. In contrast, Sloleks was constructed purely on the basis of quantitative criteria, i.e. it includes 100,000 lemmas which had the highest frequency in the large 600-million-word corpus FidaPLUS, and the majority of tokens occurring in the ssj500k corpus, which also explains its lower OOV rates in Figure 1. In any case, the different principles in the creation of the lexicons account for the larger size of hrLex lexicon and also explain the large difference in the ambiguity of the two lexicons, as possessive adjectives and proper nouns have a somewhat higher ambiguity than the remainder of the lexicon: possessive adjectives have an ambiguity of 4.68, and proper nouns of 3.78.

3 Experiment setup

3.1 Tool

We perform experiments on morphosyntactic tagging, lemmatization and dependency parsing via the `stanfordnlp` tool, one of the best-performing systems in the CoNLL shared task in 2018 (Zeman et al., 2018) with code released and a vivid development community.[6] The details on the implementation of the tool are given in (Qi et al., 2018). The tool assumes that morphosyntactic tagging is performed first, producing the UPOS and FEATS annotation layer. Next, lemmatization is performed by using the UPOS (but not FEATS) predictions. Finally, parsing is performed by exploiting all previously predicted layers (UPOS, FEATS and LEMMA). We investigate the impact of additional data on all three tasks by (1) retraining morphosyntactic tagging and lemmatization models with more data and (2) performing a simple intervention in the lemmatization procedure so that the lexicon lookup is not performed over the training data only, but the external inflectional lexicon as well.

[6] `https://github.com/stanfordnlp/stanfordnlp`

3.2 Data split

The `babushka-bench`[7] is a benchmarking platform currently used for three South Slavic languages, namely Slovenian, Croatian and Serbian (Ljubešić and Dobrovoljc, 2019). The name of the benchmarking platform comes from the idea that similar splits of data may be performed for various levels of linguistic annotation in a dataset, regardless of the fact that not all data is annotated on all linguistic levels. Each dataset is split (with a fixed random seed) via a pseudorandom function so that 80% of the data is allocated for train, while 10% is allocated to dev and 10% to test. If the dataset is split on a linguistic level which is not covered in the whole dataset, instances that do not have that level of annotation are simply discarded. What such a split enables, which becomes evident in this research already, is that it is safe to use training data from the split on the morphosyntactic level (which is applied on the whole dataset) and use the resulting model on experiments on the dependency syntax level (which is available in less than half of the dataset) without fears of data spillage between train, dev and test (e.g. the test set for parsing containing sentences that are used for training the tagger, therefore applying the tagger on the parsing test data would produce unrealistically good tags, thereby unrealistically improve parsing). The size of the data splits, both on the morphosyntactic (i.e. UD morphology annotations) and the dependency syntax (i.e. full UD annotations) levels, used in this research is given in Table 3.[8]

	ssj500k	**sl UD**	**hr500k**	**hr UD**
train	474,322	110,711	415,328	165,989
dev	62,967	16,589	39,765	14,184
test	48,959	13,370	51,364	16,855
Σ	586,248	140,670	506,457	197,028

Table 3: The benchmarking data split of the ssj500k and hr500k corpora and their officially released UD subsets.

3.3 Training and evaluation

The experiments in this paper are organised in two parts: the experiments with an extended training corpus on the level of morphosyntax and lemma and the experiments on adding an inflectional lexicon to the lemmatization process.[9]

While we perform experiments on the levels of morphosyntax, lemma and dependency syntax, we use gold segmentation to simplify our experiments as different tokenisers and sentence splitters are available for the two languages in question. Performing different preprocessing on the two languages would blur our experiments. On the other hand, applying the out-of-the box segmentation of `stanfordnlp` would produce results that are detrimental to those of our rule-based tokenizers and sentence splitters.[10] Overall, our previous experiments show that true segmentation deteriorates the results slightly on all levels of annotation, but that relations between results of different systems or setups hold regardless of whether gold or true segmentation is used.

When performing training and evaluation on levels of lemmatization and dependency syntax, we pre-annotate all the three data portions (train, dev and test) with the models from the upstream levels. We therefore apply morphosyntactic models on the data to be used for training and evaluating lemmatization, and we apply morphosyntactic tagging and lemmatization before training and evaluating dependency parsing models. While it is to be expected that training and applying the models on the training data will give an

[7] `https://github.com/clarinsi/babushka-bench`

[8] For both languages, the babushka split of data with full UD annotations differs from the official UD data releases, which are advised not to change across UD releases. However, baseline experiment results for both data split versions remain comparable (see Section 4).

[9] We do not consider improving morphosyntactic annotation via the inflectional lexicon in this paper as initial experiments have shown that various approaches to simple application of the inflectional lexicon (via lookup) do not yield any improvements. Exploiting the inflectional lexicon, probably while training the morphosyntactic annotation model, is left for future work.

[10] Readers interested in the comparison between the various segmenters should investigate the results published at `https://github.com/clarinsi/babushka-bench`

unrealistically good automatic annotation of the training data, our intuition is that, given that development data can be considered realistically annotated, the final impact of this simplifying solution (jack-knifing, i.e. annotating the training data via cross-validation would be an alternative) on the quality of annotation of the test (or any other) data will be minimal, if any. Simply preannotating training data with the model trained on that same data was also the approach taken by the developers of `stanfordnlp` during the CoNLL 2018 shared task (Qi et al., 2018).

The experiments on using a larger dataset for training the morphosyntactic tagging and lemmatization models, for which we expect to have a positive impact on the parsing quality, are split into two main parts: (1) training and evaluating morphosyntactic tagging and lemmatization models on the UD data and on all the available data, and (2) applying both models as pre-processing for training and evaluating models for dependency parsing.

The experiments on using the inflectional lexicon for improving lemmatization by extending the lookup method on the external lexicon, consist, similarly, of the experiments on training and evaluating the lemmatization models based on UD and all the available data, both with and without the lexicon, and inspecting the impact of the improved lemmatization on the parsing quality.

We evaluate all approaches with the evaluation script of the CoNLL 2018 shared task (Zeman et al., 2018), reporting F1 on all relevant levels, these being LEMMA, UPOS, XPOS, FEATS scores for morphology. For dependency syntax, the standard unlabelled (UAS) and labelled (LAS) attachment scores are complemented with the recently proposed morphology-aware labelled attachment score (MLAS), which also takes part-of-speech tags and morphological features into account and treats function words as features of content words, and bi-lexical dependency score (BLEX), which is similar to MLAS, but also incorporates lemmatization. For evaluation, we use only the UD portions of the test datasets to keep the numbers obtained on the UD data and the extended data as comparable as possible.

4 Results

The results, summarized in Tables 4 and 5, show the improvements in baseline `stanfordnlp` lemmatization, tagging and parsing performance for Croatian and Slovenian, based on the integration of the newly available training datasets for UD tagging and lemmatization (Section 4.1) and large-scale inflectional dictionaries (Section 4.2) for lemmatization.

4.1 Training corpus

Table 4 shows that re-training the lemmatization and tagging models on larger UD training sets (the ssj500k and hr500k corpora) improves the baseline performance obtained on officially released UD data[11] for both languages and for all evaluation metrics selected. In particular, the largest improvements are observed for lemmatization (+1.56pp for Slovenian and +0.91 for Croatian), language-specific JOS MSD tagging (XPOS) (+1.35 / +0.52) and universal morphological feature prediction (+1.28 / +0.53). The impact of a threefold training set increase is much less pronounced for universal PoS categories on an absolute scale (+0.24 / +0.14), but also on a relative one (15.6% vs. 31% relative error reduction on Slovenian UPOS vs. XPOS), which shows greater benefits of additional training data for the more complex layers of detailed morphosyntactic description.

As expected, retraining the parsing models on data with improved (predicted) morphology, benefits the parsing performance, as well, esp. for the morphology-sensitive scores MLAS (+1.98 / +1.34) and BLEX (+2.92 / +2.11). For the standard LAS score, the improvements amount to approx. 0.7pp for both languages. For all selected metrics, the improvements for Slovenian data are higher in comparison to Croatian, which is understandable given the differences in training data increase, i.e. a 4.3-fold increase for Slovenian and a 2.5-fold increase for Croatian (Figure 3).

[11] The baseline `stanfordnlp` performance on `babushka-bench` split is similar to that on official splits, as reported in `https://stanfordnlp.github.io/stanfordnlp/performance.html`, with the exception of FEATS prediction for Croatian, where official UD data has a specifically hard test set in comparison to the training data.

	sl UD	sl 500	hr UD	hr 500
LEMMA	95.88	97.44	95.30	96.21
UPOS	98.45	98.69	97.91	98.05
XPOS	95.65	97.00	94.60	95.12
FEATS	95.95	97.23	95.13	95.66
UAS	93.40	93.72	90.22	90.76
LAS	91.62	92.28	85.30	86.00
MLAS	84.24	86.22	75.54	76.88
BLEX	84.04	86.96	76.45	78.56

Table 4: Improvements in baseline `stanfordnlp` lemmatization, tagging and parsing performance for Croatian and Slovenian through a larger training set for UD morphology.

4.2 Lexicon of inflected forms

The results in Table 5 show that introducing large-scale morphological dictionaries (the Sloleks and hrLex lexicons) through a simple dictionary lookup significantly improves the performance of the lemmatization models trained on official UD training data alone (+2.6pp for Slovenian / +1.94 for Croatian). Noticeable improvements are also observed in comparison to the lemmatization models trained on the two larger training sets (+1.45 / +1.08), illustrating the overall benefits of morphological dictionaries in lemmatizing morphologically rich languages. As expected, the improvements are higher for Slovenian, given the larger number of OOV tokens in hr500k in comparison to ssj500k (Table 1).

Nevertheless, with the exception of lemmatization-aware BLEX score (+2.05 / +1.45), the gains in lemmatization just mildly transfer to parsing scores, giving small or no improvements. Interestingly, the improvements of parsing by improving lemmatization are consistent and overall more visible when morphosyntax and lemmatization are not improved with the extended training corpus (columns sl UD (+lex) and hr UD (+lex)), showing a saturation effect when both new datasets are exploited.

	sl UD	+ lex	sl 500	+ lex	hr UD	+lex	hr 500	+ lex
LEMMA	95.88	98.48	97.44	98.89	95.30	97.24	96.21	97.29
UAS	93.40	93.43	93.72	93.72	90.22	90.53	90.76	90.44
LAS	91.62	91.75	92.28	92.27	85.30	85.81	86.00	85.85
MLAS	84.24	84.34	86.22	86.05	75.54	76.16	76.88	76.83
BLEX	84.04	88.00	86.96	89.01	76.45	79.60	78.56	80.04

Table 5: Improvements in baseline `stanfordnlp` lemmatization, tagging and parsing performance for Croatian and Slovenian through a simple lexicon lookup for lemmatization.

5 Conclusion

This paper presented the development and the content of newly available UD-compliant training corpora and inflectional dictionaries for Slovenian and Croatian morphology processing, and illustrated their potential value to state-of-the-art tools for UD processing. Specifically, our results show that both types of resources substantially improve the baseline lemmatization, PoS tagging and morphological feature prediction performance obtained on officially released UD datasets for each language, contributing to slight improvements in dependency parsing performance, as well.

These results give important insight into the possible improvements of future text-processing tools for Slovenian, Croatian and other morphologically rich languages, where large-scale manually annotated corpora and morphological dictionaries remain relevant resources in neural-based architectures, as well. At the same time, they also raise a general question for the wider community on the optimal format, distribution and documentation of such satellite UD resources, which are likely to exist in many different languages or can eventually emerge from other similar initiatives aimed at cross-lingual annotation of specific linguistic layers (Kirov et al., 2016; Petrov et al., 2012).

Acknowledgements

The authors acknowledge the financial support from the Slovenian Research Agency through the research core funding no. P6-0411 (*Language resources and technologies for Slovene language*), the research project no. J6-8256 (*New grammar of contemporary standard Slovene: sources and methods*) and the Slovenian research infrastructure CLARIN.SI.

References

Željko Agić and Nikola Ljubešić. 2015. Universal Dependencies for Croatian (that work for Serbian, too). In *The 5th Workshop on Balto-Slavic Natural Language Processing*, pages 1–8, Hissar, Bulgaria. INCOMA Ltd. Shoumen, Bulgaria.

Kaja Dobrovoljc, Tomaž Erjavec, and Simon Krek. 2017a. The Universal Dependencies Treebank for Slovenian. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing, BSNLP@EACL 2017*, pages 33–38.

Kaja Dobrovoljc, Simon Krek, and Tomaž Erjavec. 2017b. The Sloleks Morphological Lexicon and its Future Development. In Vojko Gorjanc, Polona Gantar, Izok Kosem, and Simon Krek, editors, *Dictionary of Modern Slovene: Problems and Solutions*, pages 42–63. Ljubljana University Press: Faculty of Arts.

Kaja Dobrovoljc, Simon Krek, Peter Holozan, Tomaž Erjavec, and Miro Romih. 2015. *Morphological lexicon Sloleks 1.2*. Slovenian language resource repository CLARIN.SI. `http://hdl.handle.net/11356/1039`.

Kaja Dobrovoljc, Simon Krek, Peter Holozan, Tomaž Erjavec, Miro Romih, Špela Arhar Holdt, Jaka Čibej, Luka Krsnik, and Marko Robnik-Šikonja. 2019. *Morphological lexicon Sloleks 2.0*. Slovenian language resource repository CLARIN.SI. `http://hdl.handle.net/11356/1230`.

Tomaž Erjavec, Darja Fišer, Simon Krek, and Nina Ledinek. 2010. The JOS Linguistically Tagged Corpus of Slovene. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Tomaž Erjavec. 2012. MULTEXT-East: morphosyntactic resources for Central and Eastern European languages. *Language Resources and Evaluation*, 46(1):131–142.

Tomaž Erjavec and Simon Krek. 2008. The JOS morphosyntactically tagged corpus of Slovene. In *LREC 2008*.

Miha Grčar, Simon Krek, and Kaja Dobrovoljc. 2012. Obeliks: statistični oblikoskladenjski označevalnik in lematizator za slovenski jezik. (Obeliks: statistical morphosyntactic tagger and lemmatizer for Slovene). In *Proceedings of the 8th Language Technologies Conference*, volume C, pages 89–94, Ljubljana, Slovenia. IJS.

Christo Kirov, John Sylak-Glassman, Roger Que, and David Yarowsky. 2016. Very-large scale parsing and normalization of wiktionary morphological paradigms. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Simon Krek, Kaja Dobrovoljc, Tomaž Erjavec, Sara Može, Nina Ledinek, Nanika Holz, Katja Zupan, Polona Gantar, Taja Kuzman, Jaka Čibej, Špela Arhar Holdt, Teja Kavčič, Iza Škrjanec, Dafne Marko, Lucija Jezeršek, and Anja Zajc. 2019. *Training corpus ssj500k 2.2*. Slovenian language resource repository CLARIN.SI. `http://hdl.handle.net/11356/1210`.

Simon Krek, Kaja Dobrovoljc, Tomaž Erjavec, Sara Može, Nina Ledinek, Nanika Holz, Katja Zupan, Polona Gantar, Taja Kuzman, Jaka Čibej, Špela Arhar Holdt, Teja Kavčič, Iza Škrjanec, Dafne Marko, Lucija Jezeršek, and Anja Zajc. 2018. *Training corpus ssj500k 2.1*. Slovenian language resource repository CLARIN.SI. `http://hdl.handle.net/11356/1181`.

Nikola Ljubešić and Filip Klubička. 2016. *Croatian web corpus hrWaC 2.1*. Slovenian language resource repository CLARIN.SI. `http://hdl.handle.net/11356/1064`.

Nikola Ljubešić. 2019. *Inflectional lexicon hrLex 1.3*. Slovenian language resource repository CLARIN.SI. `http://hdl.handle.net/11356/1232`.

Nikola Ljubešić, Željko Agić, Filip Klubička, Vuk Batanović, and Tomaž Erjavec. 2018a. hr500k–a reference training corpus of croatian. In *Proceedings of the Conference on Language Technologies & Digital Humanities 2018 (JT - DH 2018)*, pages 154–161, Ljubljana, Slovenia.

Nikola Ljubešić, Željko Agić, Filip Klubička, Vuk Batanović, and Tomaž Erjavec. 2018b. *Training corpus hr500k 1.0*. Slovenian language resource repository CLARIN.SI. `http://hdl.handle.net/11356/1183`.

Nikola Ljubešić and Kaja Dobrovoljc. 2019. What does Neural Bring? Analysing Improvements in Morphosyntactic Annotation and Lemmatisation of Slovenian, Croatian and Serbian. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing, BSNLP@ACL 2019*.

Nikola Ljubešić and Tomaž Erjavec. 2016. Corpus vs. Lexicon Supervision in Morphosyntactic Tagging: the Case of Slovene. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Nikola Ljubešić, Filip Klubička, Željko Agić, and Ivo-Pavao Jazbec. 2016. New Inflectional Lexicons and Training Corpora for Improved Morphosyntactic Annotation of Croatian and Serbian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal Dependency Parsing from Scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman et al. 2017. CoNLL 2017 Shared Task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Universal Dependencies in a galaxy far, far away...

What makes Yoda's English truly alien

Natalia Levshina
Leipzig University
IPF 141199 Nikolaistraße 8-10
04109 Leipzig
natalia.levshina@uni-leipzig.de

Abstract

This paper investigates the word order used by Yoda, a character from the Star Wars universe. His clauses typically contain an Object, Oblique and/or non-finite part of the predicate followed by the subject and the finite predicate/auxiliary/copula, e.g. *Help you it will*. Using the sentences in Yodish from the scripts of the Star War films, this paper examines three cross-linguistically common tendencies, which can be explained by optimization of processing: the trade-off between entropy of S and O order and morphological cues, minimization of dependency lengths, and the tendency to place the verb in the end of a clause. For comparison, a standardized version of Yoda's sentences is used, as well as the Universal Dependencies corpora. The results of quantitative analyses indicate that Yodish is less adjusted to human processor's needs than standard English and other human languages.

1 Introduction

It is well-known that some word order patterns are more likely to occur than others. For example, the overwhelming majority of the world's languages tend to place subject before object (Tomlin 1986; Dryer 2013). There are also more subtle tendencies, which become visible only with the help of quantitative analyses. For instance, it has been shown that language users tend to minimize the average syntactic dependency lengths (Futrell et al. 2015). These and other tendencies have been explained by optimization of processing and human cognitive biases. In particular, the dependency length minimization has been explained by the limitations of human working memory (Gibson 1998; Hawkins 2004; Futrell et al. 2015), whereas the preference for placing subject before object has to do with the semantic and pragmatic properties of the main arguments, and the fact that topical and animate arguments, which are usually subjects, are more likely to appear first (Tomlin 1986).

These explanations are based on specific assumptions about human cognition. But what if the language user is an alien? The present study investigates the properties of word order observed in the speech of Yoda, a powerful Master Jedi from the Star Wars universe. Yoda appeared in most of the films of the franchise (Episodes I, II, III, V and VI, as well as the sequels *The Force Awakens* and *The Last Jedi*, as a voice). Yoda belongs to an unknown species. One of his distinctive characteristics, in addition to large green ears, is the use of unusual word order patterns. Some examples are provided below:

(1) Friends you have there. (E V)[1]
(2) Help you it will. (E II)
(3) The secret of the Ancient Order of the Whills, he studied. (E III)

[1] 'E' stands for Episode.

Yoda's version of English will be called Yodish in this study. But why should a study of an artificial language like Yodish be interesting to syntacticians? These data allow us to understand better the properties of human languages, in particular, what is possible and impossible, likely and unlikely, efficient and inefficient. In particular, we will focus on the following manifestations of word order efficiency:
1) the implicational relationship between lack of rich morphology (especially case system and agreement) and rigid word order (e.g. Sapir 1921; Kiparsky 1997);
2) the universal tendency towards minimization of dependency lengths (Hawkins 2004; Futrell et al. 2015);
3) the preference for the final position of the verb in a clause. This tendency has been explained by the efficient strategy of putting verbs, which require more cognitive efforts in terms of categorization and acquisition than nouns, in the position where the former are more predictable and therefore easier to process (Ferrer-i-Cancho 2017).

In order to investigate whether these cross-linguistic tendencies are also observed in Yodish, we use the film scripts of five Star Wars films parsed according to the Universal Dependencies annotation schema. These data are compared with the Universal Dependencies corpora (Nivre et al. 2017) and with a standardized version of Yodish, created manually by myself.

The remaining part of the paper is organized as follows. Section 2 presents the data and describe the properties of Yoda's word order in Section 2. Next, Yodish is compared with human languages, using the Universal Dependencies corpora and the normalized version of Yoda's sentences in standard English. In particular, the implication relationship between rich morphology and rigid word order is discussed in Section 3, followed by a study related to minimization of dependency lengths (Section 4), and the position of verb in a clause (Section 5). Section 6 provides a summary and a discussion of the findings.

2 Yodish: data and main word patterns

The Yodish data have been collected from the Internet Movie Scripts Database.[2] Data from five episodes were used: two episodes from the original trilogy (Episodes V and VI) and three episodes from the prequel trilogy (Episodes I, II and III). One-word utterances and paratactic structures were excluded as irrelevant for the study. In total, the sample contained approximately 2000 words.

Yoda's word order has been described as some linguists as OSV or XSV, where X stands for any complement that goes with the verb.[3] However, it seems that a more precise description would be as follows:

(4) Non-finite part of predicate/Object/Oblique Subject Finite Verb/Auxiliary/Copula

The first part can be object, oblique, nominal part of the predicate or non-finite parts of the predicate, i.e. participle or infinitive with dependent elements. They are followed by the subject and the finite verb, auxiliary or copula. Below are some examples that support this interpretation:

(5) Rest I need (E VI)
(6) To his family, send him. (E III)
(7) A certainty it is. (E II)
(8) Hard to see, the dark side is. (E I)
(9) Earned it, I have (E VI)

If the non-finite lexical verb has arguments, they usually follow the verb, as in (10):

(10) **Save them**, we must. (E III)

[2] See https://www.imsdb.com/ (last access 29.04.2019).

[3] See a blog post by Geoffrey K. Pullum at http://itre.cis.upenn.edu/~myl/languagelog/archives/002173.html, a post by Mark Libermann at http://itre.cis.upenn.edu/~myl/languagelog/archives/002182.html, and an article in The Atlantic by Adrienne Lafrance https://www.theatlantic.com/entertainment/archive/2015/12/hmmmmm/420798/ (last access 29.04.2019).

But occasionally an argument may precede the verb, as in (11):

(11) Master Kenobi, **our spies contact**, you must, and then wait. (E III)

The subject is usually followed by the finite form, including copulas and auxiliaries, as in (12), although some sentences have the reverse order: first copula/auxiliary, and then subject, as in (13):

(12) Subject – copula/AUX:
Unexpected **this is**, and unfortunate. (E VI)
Earned it, **I have** (E VI)

(13) Copula/AUX – subject:
Not ready for the burden **were you**. (E VI)
Heard from no one, **have we**. (E III)

In addition, there are rare cases when the non-finite verb is in the end, after the modal or auxiliary:

(14) The outlying systems, you must **sweep**. (E III)

The most substantial source of variation, however, is the frequent use of conventional SVO word order, as in the examples below:

(15) Master Obi-Wan has lost a planet. (E II)
(16) A Jedi's strength flows from the Force. (E III)
(17) That place is strong with the dark side of the force. (E V)

Compare two very similar sentences with different word order:

(18) Reckless is he. (E V)
(19) You are reckless (E V)

It is not clear what may condition this variation. In the absence of evidence to the contrary, we assume that Yodish is a special variety and does not represent code-switching between standard English and Yoda's own dialect.

One should mention that OSV is not entirely alien to English speakers: it is permissible in certain emphatic contexts, as the one below:

(20) **This** I could **understand** but why not tell me this some three and a half hours earlier.[4]

However, the structures in Yodish are not emphatic or contrastive like (20).

Remarkably, the films vary substantially in terms of the deviations of Yoda's word order from standard English. In order to quantify the differences, the original (Yoda's) token IDs were compared with the IDs in the normalized version, re-written in standard English, and the proportions of discrepancies for each of the five films were computed. The original trilogy has lower number of words that deviate in their order from standard English: Episode V has only 25%, while Episode VI has 36.1%. The prequel films have higher numbers. In Episode I, the difference is 56.3%; Episode II has 51.2%. Finally, Episode III has 67.4%, the highest number of deviations.

[4] URL https://www.nhs.uk/Services/hospitals/ReviewsAndRatings/DefaultView.aspx?id=95110&SortType=1,5,5,5,5,5,5&pageno=3&subject=All%20subjects&spid=0 (last access 29.04.2019).

It is interesting that the events of Episode V happen after Yoda was in exile for many years. Therefore, one would expect that his solitude would lead to less standard English, due to the lack of language contact. However, the opposite is the case. This counterintuitive fact can be explained by Yoda's character development, which does not correspond to the timeline in the Star Wars universe. First, the original Episodes V and VI were created, and only later the prequel films with Episodes I to III. As Yoda's film character was developed by the authors (primarily, by George Lucas), the divergent word order became more and more frequent.

3 Trade-off between word order entropy and morphology

Recognition of the core arguments, or who did what to whom, is crucial for interpretation of a sentence. Rigid word order can compensate for the lack of morphological cues that help to recognize the main arguments, as in English or Mandarin Chinese. Flexible word order is observed overwhelmingly in languages with rich morphology, such as the Slavic or Baltic languages. This relationship, which is well-known in linguistics (Sapir 1921: 66; Jakobson 1936[1971]: 28; Blake 2001: 15), has been formulated as a one-way implication by Kiparsky: "lack of inflectional morphology implies fixed order of direct nominal arguments" (Kiparsky 1997: 461; see also McFadden 2003: 301). Similar claims have been supported by typological evidence (Siewierska 1998; Sinnemäki 2008), corpus data (Futrell et al. 2015) and experimental evidence (Fedzechkina et al. 2016).

Yodish, like English, has poor inflectional morphology and no case distinctions for nominal subject and object. Therefore, it must have a strong preference for a specific ordering of subject and object. In order to see if this prediction holds, Shannon's entropy was computed using the standard formula:

(21) $H(X) = -\sum_{i=1}^{2} P(x_i)\, log_2\, P(x_i)$

where X is a binary variable representing two possible word orders, Subject + Object (SO) and Object + Subject (OS). P (x_i) is the probability of one of the orders, which equals its relative frequency (proportion) in the corpus. If the proportion of SO is 1, and the proportion of the reverse order OS is 0, or the other way round, the entropy H is equal to zero. That is, there is no variation. If the proportion of each of the possible word orders is 0.5, the entropy takes the maximum value of 1. If both orders are attested, and one of them is more frequent than the other, then H lies between 0 and 1.

The frequencies of SO and OS were collected manually, since the number of sentences with an explicit nominal or pronominal subject and object is rather low. The results for each film are shown in Table 1. Only the main clauses were taken into account because the order of subordinators (complementizers and relativizers) is determined by other syntactic factors. Non-finite clauses and questions were excluded, as well. The entropy values are high. One can see that there is no strongly preferred order. Even though some of the frequencies are rather low, the overall pattern is quite consistent. The highest entropy is observed in Episode V, the first film where Yoda appears, and the lowest in Episode I (which may be due to the small number of examples).

Film	Frequency SO	Frequency OS	H
Episode I	2	7	0.76
Episode II	3	8	0.85
Episode III	7	18	0.86
Episode V	18	15	0.99
Episode VI	3	6	0.92
Total	33	54	0.96

Table 1: SO and OS frequencies in Yodish and their entropy.

Let us now compare these results with human languages. Figure 1 displays the entropy values of subject and object (co-dependents) in Yoda's language and in the languages of the Universal Dependencies corpora (version 2.2). Only the main clauses are considered, since the object and subject in subordinate clauses can be expressed by subordinators, which influences their position. Yodish is represented by separate values for each film, plus the value Yoda_av computed on the total frequencies in all films. The Yodish values are displayed as black dots. In the UD corpora, only the corpora with the total number of subject – object pairs above 50 were considered.

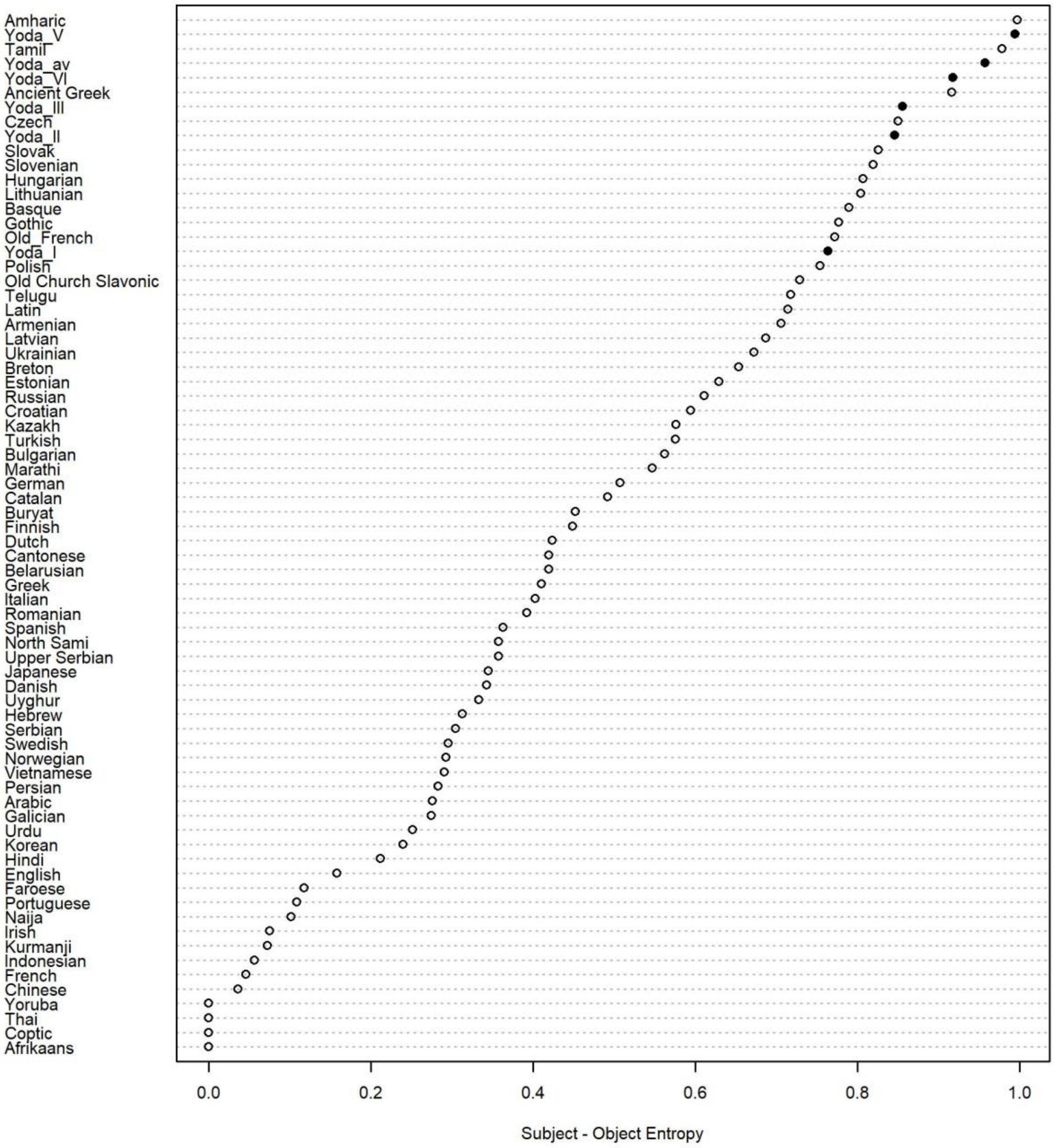


Figure 1: Subject-Object Entropy in Yodish and UD corpora.

According to the data, the highest entropy is observed in Amharic. However, this result seems to be an artefact of some annotation decisions made by the corpus creators. Amharic has subject clitics, which follow the verb to mark agreement. An inspection of a corpus sample reveals that the clitics are annotated in different ways depending on the presence of a full nominal subject. If there is no full nominal subject, they have the tag of subject ('nsubj'). In the presence of a nominal subject, they are annotated as expletive elements ('expl') – i.e. nominals that appear in an argument position of a predicate but which do not themselves satisfy any of the semantic roles of the predicate. This explains the high variability of the subject and object word order in Amharic.

The next lines are occupied by some variants of Yodish, with Tamil, Ancient Greek and some Slavic, Baltic and other synthetic languages in-between. Thus, Yodish behaves like the languages that have rich morphology and case marking. In contrast, English and other analytic languages are located at the bottom because they have low entropy values. From this follows that Yodish is truly alien. The lack of syntactic or morphological cues means that the hearer needs to invest extra processing efforts in order to understand who did what to whom.

4 Dependency lengths

This section addresses the question whether Yodish is efficient with regard to dependency lengths. As was already mentioned in Section 1, human languages minimize the distance between the heads and dependents. This has been explained in terms of processing efficiency: long dependencies are more difficult to process. When the head and dependent are located far away, it creates extra load for memory because of the long timespan over which the head or the dependent must be held in a memory store (Hawkins 2004; Futrell et al. 2015).

It does not seem very reasonable to compare dependency lengths across different languages because the number of words depends on the properties of a language (e.g. synthetic or analytic) and orthographic conventions. In order to have a basis for comparison, the Yodish data were normalized such that the order was like in standard English. Sentences that were difficult to transform into normal English were excluded. So were one-word replies and parataxic structures. In total, the dataset had 305 sentences in both orders.

Next, the sentences were parsed syntactically with the help of the R package udpipe (Wijffels 2018) according to the Universal Dependencies annotation style based on the UDPipe software (Strakov and Straková 2017). A manual check of each sentence was performed, and the parsing errors were corrected. After that, dependency lengths were computed for the normalized version and the original Yoda's version with the help of an R script. The punctuation marks were not taken into account in the computation of lengths, i.e. they were not regarded as full-fledged tokens with their own ID. Consider an example:

(22) a. **Heard** from no one, **have** we. (E III, original)
b. We **have heard** from no one. (standardized)

If we take the relationship between the auxiliary have and the participle heard, the distance in the original (22a) is 4 words (the comma is excluded). In the standardized version (22b) it is only 1 word.

It is necessary to mention here that there exist different approaches to deciding on what should be regarded as the head and what as the dependent. For example, many syntactic theories say that prepositional objects depend on their prepositions, whereas the UD corpora annotate prepositions as dependents of head nouns, marking the former as a case relation. Futrell et al. (2015) compared different approaches are concluded that the results were the same, as far as the tendency to minimize dependency lengths is concerned.

The dependency lengths of the terms of address, coordinated and parataxic sentence parts, as well as compounds (mostly nominals) were excluded from the subsequent aggregate analyses because the notion of head is difficult to apply there. The roots, which have no lexical head, were excluded, as well.

The quantitative analyses reveal that Yodish has on average longer mean dependencies than standard English. Consider the average lengths presented in Table 2. The smallest difference is in Episodes V and VI – this is due to their more standard character in general, as was already mentioned. It is interesting that the other word order is observed mostly in short sentences. In longer sentences, there is no difference. This makes the difference in dependence lengths less observable.

A series of the Wilcoxon's paired rank-sum tests (where two measures are provided for each word in the text) performed on each film demonstrate that the difference is only statistically significant in Episodes I and III, and marginally significant in Episode II. In the original trilogy, the difference is not statistically significant.

Note that Episode VI, in which Yoda dies, only has 29 analyzable sentences, so the test may not have sufficient power due to the small sample size.

Film	number of comparable sentences	mean length in original Yodish	mean length in standardized version	p-value Wilcoxon
Episode I	42	2.25	1.82	0.001
Episode II	44	2.02	1.89	0.099
Episode III	101	2.24	1.94	< 0.0001
Episode V	89	2	1.98	0.585
Episode VI	29	2.11	2.03	0.365

Table 2: Mean dependency length in Yodish (original and standardized).

Figure 2 displays the frequencies of different lengths in original Yodish and in the standardized version from all five films. The minimum dependency length is 1 – where the head is located immediately before or after the dependent. The maximum was 19, due to a very long sentence. The plot shows that the short dependencies (1 and 2 words) slightly predominate in the standardized version, but the longer ones tend to be more frequent in original Yodish.

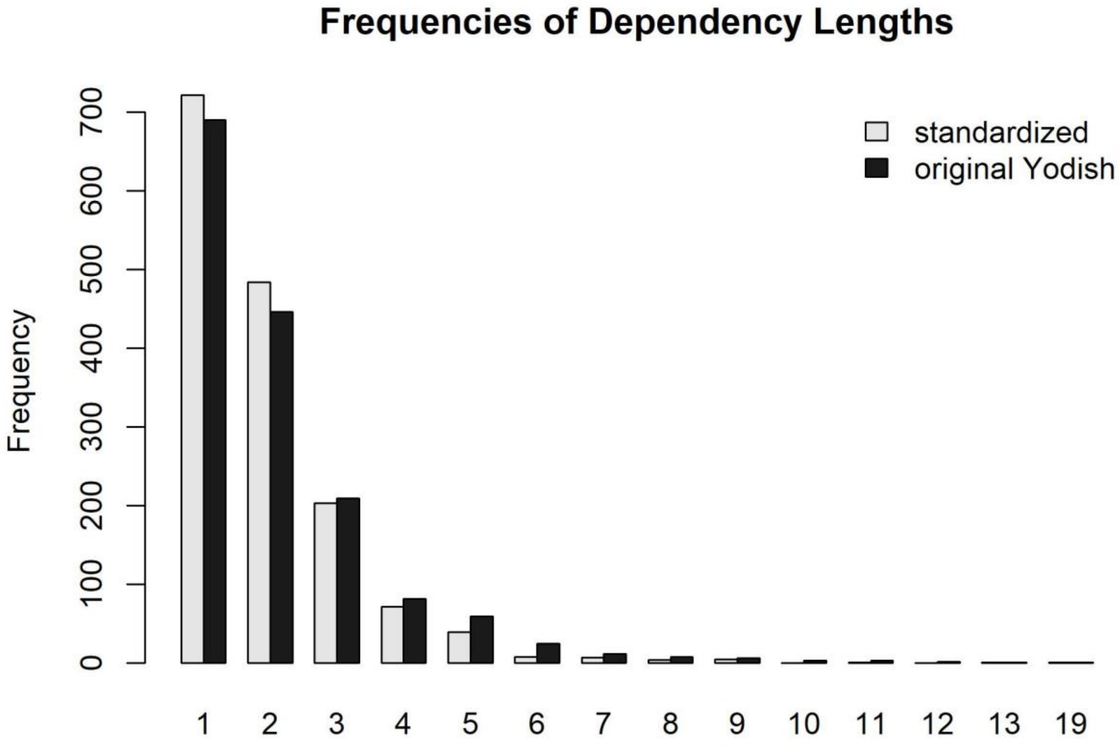


Figure 2: Frequency of dependency lengths in standardized and original Yodish.

In order to interpret these differences, it may be instructive to look at the individual dependencies. The mean lengths were computed for each dependency that occurs 3 times and more. Interestingly, the elements of a nominal phrase have the same mean dependency lengths in the original and standardized versions. These are prepositions, adjectives, nominal modifiers, determiners and numerals. Many dependencies are on average longer in the original Yodish version: auxiliary and modal verbs ('aux'), copulas ('cop'), subject and complement clauses ('csubj' and 'ccomp'), non-finite complements ('xcomp') and adverbial modifiers ('advmod'). Since the auxiliaries and copulas are the most frequent categories among those, the loss of processing efficiency is mostly due to the fact that the auxiliaries and copulas are often separated from the non-finite and nominal parts of the predicates. Below is an example:

(23) a. **Failed** to stop the Sith Lord, I **have**. (E III, original)
b. I **have failed** to stop the Sith Lord. (standardized)

In the original version (23a), the distance is 7 words, while in the standardized version (23b) it is only 1.

At the same time, some dependencies become shorter: passive subjects ('nsubj:pass'), attributive/relative clauses ('acl:relcl') and adverbial clauses ('advcl'). But they are too rare to influence the general picture substantially.

Objects, which were prominent in the previous discussions of Yodish, do not influence the general picture greatly because the move of the object to the beginning of the sentence is compensated by the shorter distance due to the presence of determiners and various NP modifiers (adjectives, etc.) before the object noun. Consider an example:

(24) a. The outlying **systems**, you must **sweep**. (E III, original)
b. You must **sweep** the outlying **systems**. (standardized)

In the original version (24a), the distance between the object *systems* and its head *sweep* is 3 words. In the normalized order (24b), the order is also 3 words. The difference is neutralized due to the presence of the determiner and adjective before the noun.

To summarize, Yodish is less efficient than standard English due to the separation of non-finite, lexical parts from the auxiliaries and copulas. This pattern is not only inefficient, it is also quite unrealistic. The reason is that grammaticalized elements, such as auxiliary verbs, arise in highly predictable contexts. For instance, the future marker *going to* is phonologically reduced (cf. *gonna*) and semantically bleached only in the contexts where it is followed by a verb. When the auxiliary is not accompanied by the lexical part, it is less predictable and therefore less likely to undergo phonological reduction and semantic changes. This frequent co-occurrence of the elements together, which is necessary for grammaticalization, explains why grammaticalized units display formal bondedness with the lexical elements (Lehmann 2015: Section 4.3.2). Moreover, auxiliaries usually lose their positional freedom (*Ibid*.: 168). The existence of auxiliaries in Yodish, which are often split from their lexical elements – which is probably their position in Yoda's native language – is then difficult to explain. This is another piece of evidence showing that Yodish is truly alien.

5 Position of verb in the clause

Most languages of the world place verb in the end of the clause. Ferrer-i-Cancho (2017) argues that this order is efficient because verbs represent cognitive difficulties for language users and learners. For children, verbs are more difficult to learn; actions are harder to verbalize and recall. Non-verbal experiments show a robust strong preference for an order consistent with subject-object-verb even in speakers whose language has a different dominant word order. Therefore, it is efficient to facilitate the processing and the learning of the most difficult item, i.e. verb, by minimizing the uncertainty about the verb. Note that this principle is in competition with dependency length minimization. In particular, SVO languages are also quite common because this order minimizes the distances between the head (the verb) and the dependent arguments.

As was already mentioned, Yodish tends to put the finite verbs (auxiliaries and copulas) last more often than normal English. Is it then more efficient than English because the verbs are more predictable? This is unlikely. It is the lexical verb that should be predicted from the arguments, rather than an auxiliary, which expresses the abstract temporal, aspectual and modal properties of the proposition. However, the lexical verb, as the non-finite part of the predicate, often precedes the arguments. How much does it influence the efficiency of Yodish?

Table 3 displays the frequencies of lexical verbal predicates (no auxiliaries or copulas) in three conditions: a) the verb precedes all other arguments (minimum predictability of the verb); b) one argument (S or O) precedes the verb (medium predictability) and c) two arguments (S and O) precede the verb (maximal predictability). When obtaining the counts, the questions, passives and sentences with clausal complements were omitted. Only the main clauses were analyzed.

While the standardized version clearly prefers medium predictability (due to the default SVO and SV patterns), there are clearly more cases in Yodish both with minimal predictability (58 against 21) and maximal predictability (23 against 0). All examples of the latter type have OSV order:

(25) Grave danger I fear in his training. (E I)
(26) The outlying systems, you must sweep. (E III)
(27) I hear a new apprentice, you have. (E III)

Version	Zero arguments before the lexical verb, minimal predictability	One argument before the lexical verb, medium predictability	Two arguments before the lexical verb, maximal predictability
Original Yodish	58 (37.9%)	72 (47.1%)	23 (15%)
Standardized	21 (13.7%)	132 (86.3%)	0 (0%)

Table 3: Position of lexical predicate in original and standardized Yodish.

In many cases, the final lexical verb is highly frequent and abstract, or 'light', e.g. *have* or *need*, as in (27). There verbs are unlikely to present problems for processing due to their high accessibility. Moreover, the minimal predictability group has the difference of 37 in favour of Yodish, and the maximal predictability group only the difference of 23. This means that Yodish is overall less efficient than English.

6 Discussion

The present study has examined several common tendencies in human languages, which can be explained by considerations of processing efficiency, and investigated whether these tendencies can be found in Yodish.

First, human languages exhibit an implicational relationship between rigid order of subject and object and lack of rich morphology (in particular, case system and agreement). Rigid word order serves as a compensation for lack of morphological tools for expressing 'who did what to whom'. Contrary to this prediction, Yodish has very high variability of the order of subject and object, despite the fact that it has no case marking.

Second, human languages tend to minimize dependency lengths, which saves working memory from overload. Yodish has on average longer syntactic dependencies than the corresponding standard English version. The differences are particularly evident in the prequel films. Longer dependencies mean greater processing load. The main causes of longer dependencies are auxiliaries and copulas, which are separated from the non-finite and nominal predicates. One may wonder how these words have managed to grammaticalize into tense, aspect and modality markers, given that they have no immediate linguistic context to rely on. Their positional variability and lack of bondedness with the lexical part are also highly untypical of grammaticalized units in human languages. This suggests that the grammaticalization processes in Yoda's native language are very different from the ones in human language, if they exist at all.

Third, most languages tend to put verb in the final position, which improves the processing due to its higher predictability given the arguments. However, Yodish more often puts the lexical predicate in the initial position than English, which means that the processing of the verb is not facilitated by the knowledge of the main arguments.

Interestingly, as Yoda's character develops, his word order diverges from standard English. There are differences between the original trilogy and the prequel films. The average dependency length increases. At the same time, the SO and OS entropy decreases, which makes Yodish more consistent and learnable.

How can we explain these differences between human languages and Yodish? First of all, Yoda is a powerful and wise Jedi, who is in command of the Force, which is defined by Obi-Wan Kenobi, another prominent Jedi, as follows:

> ...the Force is what gives a Jedi his power. It's an energy field created by all living things. It surrounds us and penetrates us. It binds the galaxy together (Episode IV).

Thanks to the Force, it is not surprising that the processing capacity of a Jedi is also superior to those of normal humans. At the same time, Yoda does not seem to care much about the fact that his word order may be challenging for normal listeners, since he uses this order not only to speak to other Jedi, but also to normal humans and other species. It is unlikely that he cannot master standard English, or any other language. There are two possible explanations. First, due to his different cognitive make-up, he cannot estimate that this word order is suboptimal for the others. Second, Yoda acts as a dedicated teacher, trying to make his interlocutor's mind work harder, grow and develop.

Finally, let us come back to the real world and take the perspective of George Lucas and other film creators responsible for Yoda's syntax, who seem to be exploiting Levinson's M-heuristic: "What's said in an abnormal way, isn't normal"; or "Marked message indicates marked situation" (Levinson 2000). Extra efforts spent during the processing of Yoda's utterances promise the film audience extra benefits in the form of additional inferences. The story teaches us that even a small green creature with strange English due to immigrant background can be a powerful Master Jedi and a hero. As Yoda says himself, "You must unlearn what you have learned" (Episode V). Thus, although we can conclude that Yoda's word order is suboptimal in the Star Wars universe, it is perfectly efficient for the communication between the film creators and the audience.

References

Barry J. Blake. 2001. *Case*. Cambridge University Press, Cambridge, UK.

Matthew Dryer. 2013. Order of Subject, Object and Verb. In: Matthew S. Dryer and Martin Haspelmath (eds.), The World Atlas of Language Structures Online. Max Planck Institute for Evolutionary Anthropology, Leipzig. (Available online at https://wals.info/chapter/81, Accessed on 2019-04-25.)

Mariya Fedzechkina, Elissa L. Newport and T. Florian Jaeger. 2016. Balancing effort and information transmission during language acquisition: evidence from word order and case marking. *Cognitive Science*, 41(2): 416–446.

Ramon Ferrer-i-Cancho. 2017. The Placement of the Head that Maximizes Predictability. An Information Theoretic Approach. *Glottometrics,* 39: 38–71.

Richard Futrell, Kyle Mahowald and Edward Gibson. 2015. Quantifying word order freedom in dependency corpora. In: *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, 91–100, Uppsala, Sweden, August 24–26, 2015.

Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition,* 68(1):1–76.

John A. Hawkins. 1994. *A Performance Theory of Order and Constituency.* Cambridge University Press, Cambridge, UK.

Roman Jakobson. 1936[1971]. Beitrag zur allgemeinen Kasuslehre. In: Roman Jakobson, *Selected Writings. Vol. II. Word and Language*, 23–71. Mouton, The Hague/Paris.

Paul Kiparsky. 1997. The rise of positional licensing. In: Ans von Kemenade & Nigel Vincent (eds.), *Parameters of morphosyntactic change*, 460–494. Cambridge University Press, Cambridge, UK.

Christian Lehmann. 2015. *Thoughts on Grammaticalization.* 3rd edn. Language Science Press, Berlin.

Stephen Levinson. 2000. *Presumptive Meanings – The Theory of Generalized Conversational Implicature*. MIT Press, Cambridge, MA.

Thomas McFadden. 2003. On morphological case and word-order freedom. In: *Proceedings of the Twenty-Ninth Annual Meeting of the Berkeley Linguistics Society: General Session and Parasession on Phonetic Sources of Phonological Patterns: Synchronic and Diachronic Explanations*, 295-306.

Joakim Nivre, Željko Agić, Lars Ahrenberg et al. 2017. Universal dependencies 2.0 – CoNLL 2017 shared task development and test data. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University. http://hdl.handle.net/11234/1-2184. See also http://universaldependencies.org/ (last access 14.12.2017).

Edward Sapir. 1921. *Language, an introduction to the study of speech*. Harcourt, Brace and Co, New York.

Anna Siewierska. 1998. Variation in major constituent order: a global and a European perspective. In: Anna Siwerierska (ed.), *Constituent Order in the Languages of Europe*, 475–552. De Gruyter Mouton, Berlin.

Kaius Sinnemäki. 2008. Complexity trade-offs in core argument marking. In: Matti Miestamo, Kaius Sinnemäki & Fred Karlsson (eds), *Language Complexity: Typology, Contact, Change*, 67–88. John Benjamins, Amsterdam.

Milan Straka and Jana Straková. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Vancouver, Canada, August 2017.

Russel S. Tomlin. 1986. *Basic Word Order: Functional Principles*. Croom Helm, London.

Jan Wijffels. 2018. udpipe: Tokenization, Parts of Speech Tagging, Lemmatization and Dependency Parsing with the UDPipe NLP Toolkit. R package version 0.7. https://CRAN.R-project.org/package=udpipe

HDT-UD: A very large Universal Dependencies treebank for German

Emanuel Borges Völker[*] and **Maximilan Wendt**[*]
Universität Hamburg
emanuel.borges.voelker@studium.uni-hamburg.de
6mwendt@informatik.uni-hamburg.de

Felix Hennig
University of Edinburgh
mail@felixhennig.com

Arne Köhn
Saarland University
koehn@coli.uni-saarland.de

Abstract

We report on the conversion of the Hamburg Dependency Treebank (Foth et al., 2014) to Universal Dependencies. The HDT consists of more than 200.000 sentences annotated with dependency structure, making every attempt at manual conversion or manual post-processing extremely costly. The conversion employs an unranked tree transducer. This formalism allows to express transformation rules in a concise way, guarantees the well-formedness of the output and is predictable to the rule writers. Together with the release of a converted subset of the HDT spanning 3 million tokens, we release an interactive workbench for writing and refining tree transducer rules. Our conversion achieves a very high labeled accuracy with respect to a manually converted gold standard of 97.3%. Up to now, the conversion effort took about 1000 hours of work.

1 Introduction

Despite the availability of several German treebanks (TIGER (Brants et al., 2004), TüBa-D/Z (Telljohann et al., 2004), HDT (Foth et al., 2014)) and a fairly active research community, the only other larger German treebank which has been converted to Universal Dependencies is TüBa-D/Z (Çöltekin et al., 2017), consisting of 95.595 sentences (1.788k tokens). Until now, the largest German treebank distributed by the UD project was German GSD, consisting of 15.590 sentences (292k tokens). As that treebank's original annotation still stems from the pre-UD time, interesting syntactic constructs are often not annotated in accordance to the UDv2 guidelines[1]. Furthermore, the German UD guidelines themselves were often not up to date, sometimes even incomplete. Our work on converting the HDT to Universal Dependencies therefore also consisted in a large part of working out the best way to represent German dependency structures in the UD annotation schema; these decisions are encoded in the resulting treebank and – where applicable – were documented to be added to the general UD documentation.

The Hamburg Dependency Treebank is a native dependency treebank developed mainly for research in parsing; the annotation schema (Foth, 2006) was developed as part of the annotation effort. The texts in the treebank stem from heise.de, a well-known German technical website reporting about new software and hardware, technology-related politics, earnings of tech companies, inter alia. Some texts are short and formulaic, others are long editorials. The HDT has an average sentence length of 18.4 and more than 200.000 sentences are manually annotated. The text of the HDT can be distributed for academic use, the annotations are licensed under a Creative Commons share-alike license. The original treebank and its conversion to Universal Dependencies are available under `https://nats.gitlab.io/hdt/`.

We will give a rough overview of other treebanks which were converted to Universal Dependencies and of the methods which were employed, explain why our approach is different and how it works. After detailing our conversion process, we discuss general issues faced when converting a treebank to a different schema as well as specific problematic structures in our case and how we dealt with them, explain how we converted morphological features and finally evaluate the results of the conversion process. We close

[*] First two authors contributed equally.

[1] For example, names consisting of several tokens are regularly incorrectly annotated as *compound* instead of *flat* (or *flat:name*) in the German GSD treebank.

with a description of the (treebank agnostic) interactive conversion workbench developed as part of the conversion effort which enabled this large scale conversion.

2 Related Work

The Universal Dependencies (UD) project (McDonald et al., 2013) has caused many treebank maintainers to convert their treebank from their schema – often only used by this one treebank – to the UD schema. Examples can be found in Swedish (Nivre, 2014; Ahrenberg, 2015), Finnish (Pyysalo et al., 2015), Danish (Johannsen et al., 2015), Norwegian (Øvrelid and Hohle, 2016), Turkish (Sulubacak et al., 2016), Hindi (Tandon et al., 2016) and North Sámi (Tyers and Sheyanova, 2017). Most conversions rely on ad-hoc scripting of the conversion process, and a lot of manual intervention.

In some cases, a more systematic approach was taken. Pyysalo et al. (2015) based their conversion on dep2dep[2], a treebank conversion tool that allows for the definition of rules which are then converted to prolog code. Tyers and Sheyanova (2017) used XSLT, an XML tree conversion language, to build a pipeline to convert their parser output from the native schema to UD. Çöltekin et al. (2017) converted their constituency treebank with an automatic approach based on traditional head-finding heuristics. Seddah et al. (2018) converted a treebank by training a parser on a different treebank annotated with Universal Dependencies, using the original source annotations as additional features to the parser. This parser is then able to produce high quality UD annotations for other treebanks of the same language and with the same source annotation schema. For German, this approach is unlikely to work in the future as the different treebanks do not share a common source annotation schema.

Hennig and Köhn (2017) developed TrUDucer, a tool to convert dependency treebanks between different schemas based on the top-down tree transducer formalism (Maletti, 2010). This work builds on TrUDucer to convert the HDT and develop an interactive workbench for treebank conversion.

3 Converting between Dependency Schemas using Tree Transducers

We utilise tree transducers to convert dependency tree structures from the HDT schema made for German (and the HDT in particular) to the more general Universal Dependencies. The tree transducer formalism builds on the use of local context for partial conversion of subtrees based on predefined rules which are applied iteratively in a top-down fashion. Figure 1 shows an example conversion of a sentence, using a tree transducer defined by the following rules:

```
n:S() -> n:root();
n:SUBJ() -> n:nsubj();
n:DET() -> n:det();
n1:PP(n2:PN()) -> n2:obl(n1:case());
```

Each rule consists of a left-hand side and a right-hand side, where the left-hand side is a description of a structure found in the source treebank, and the right-hand side the corresponding structure in the target schema. The child nodes are listed in parentheses, each node is referred to by an identifier (`n, n1, n2`) and its dependency relation to the parent node (after a colon). Node structure can be changed arbitrarily, a common use-case is the matching of a function word and a content word and then switching their position in the tree. An example of this is the last rule given above, which is applied in the step from the second tree to the third one in Figure 1.

The structures are matched in the tree. Where to match is defined by the *conversion frontier* (Shown as orange lines in Figure 1). At the start of the tree conversion the conversion frontier is set at the root of the tree. After a rule is matched and applied, the conversion frontier is moved below the nodes that have been converted in this rule. The conversion then continues at the new conversion frontier, where again a rule is applied and the frontier moved. This way the frontier is moved down from the root to the leaves, until all nodes are converted. The rule list can contain multiple matching rules with gradually decreasing specificity, allowing to describe edge cases first and defining more generic rules later.

[2] https://github.com/TurkuNLP/dep2dep

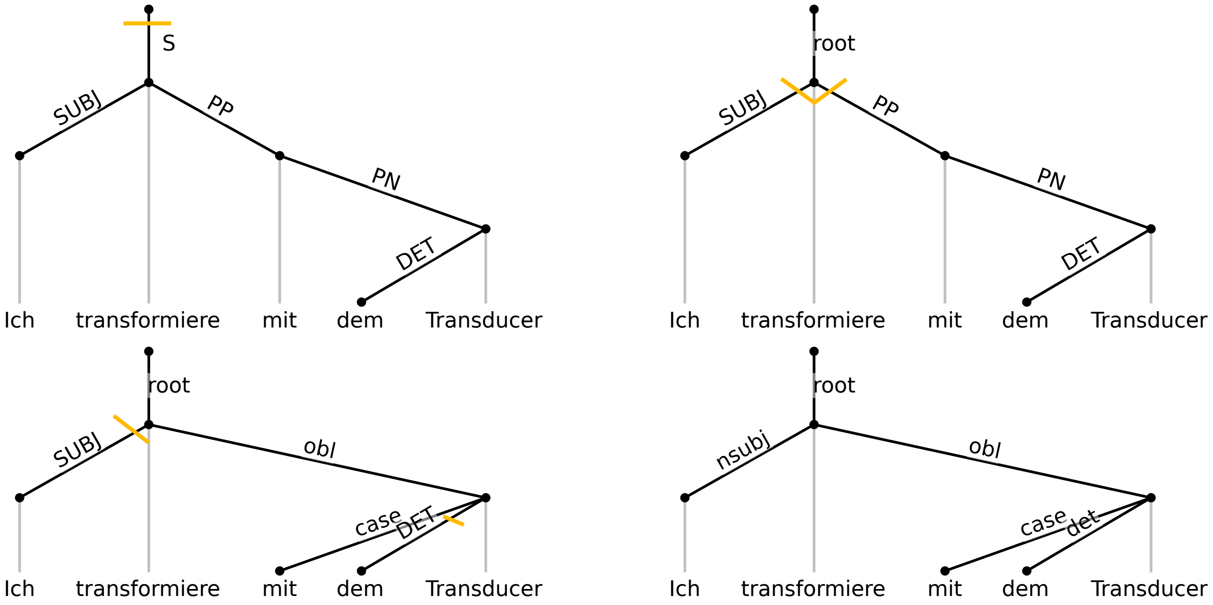


Figure 1: Conversion of "Ich transformiere mit dem Transducer" (*I transform with the transducer*); yellow lines indicate the state nodes, HDT labels are uppercase, UD are lowercase. Taken from Hennig and Köhn (2017)

This approach guarantees that trees are always well formed. As the rules all convert the tree locally, different subtrees can be converted independently of each other and different choices of where a rule should be applied next lead to the same result. This makes the process transparent to rule authors, deterministic and reproducible.

On top of this fundamental mechanism, the TrUDucer implementation provides certain convenience features. Similarly to the dependency relation, the PoS tag of a node to match can be specified directly by the use of a period. To formulate more complex conditionals with other node properties Groovy code[3] can be added to the rule in a *rule body*:

```
p.NN({n.VAFIN:AUX()}) -> p(n:cop()) :- { n.lemma == "sein" };
p:root({n:S()}) -> n:root(p:parataxis()) :- { p.ord > n.ord };
```

Groovy is a fully functional programming language and through the nodes the whole tree can be accessed and modified to accomodate edge cases. Also shown above is the use of curly braces on the left-hand side of the rule, used to match a frontier node and allowing to match more local context above the frontier. The first rule detects copula verbs (see Section 4.1), the second one checks that the node p is to the right of n. More details can be found in Hennig and Köhn (2017).

4 Converting the Hamburg Dependency Treebank to Universal Dependencies

While the HDT and UD have many similarities, most notably the use of dependency relations, they also have a few key differences which are relevant when converting the dependency trees from one schema to another. The HDT was annotated with a schema specifically tailored to the needs and particularities of the German language. In contrast, UD is a framework designed for cross-linguistic comparability. This is exemplified by the way dependency relations are treated. UD relations are headed by content words since this makes it possible to maintain high comparability when across differently structured languages[4]. HDT relations are headed by function words since this allows a more precise representation of the language-specific syntactic structure (Foth et al., 2014). The difference in focus is also noticeable when looking at what kind of information is valued and how it is represented in the respective annotation

[3] An embeddable java-based scripting language: `https://groovy-lang.org/`

[4] `https://universaldependencies.org/u/overview/syntax.html`

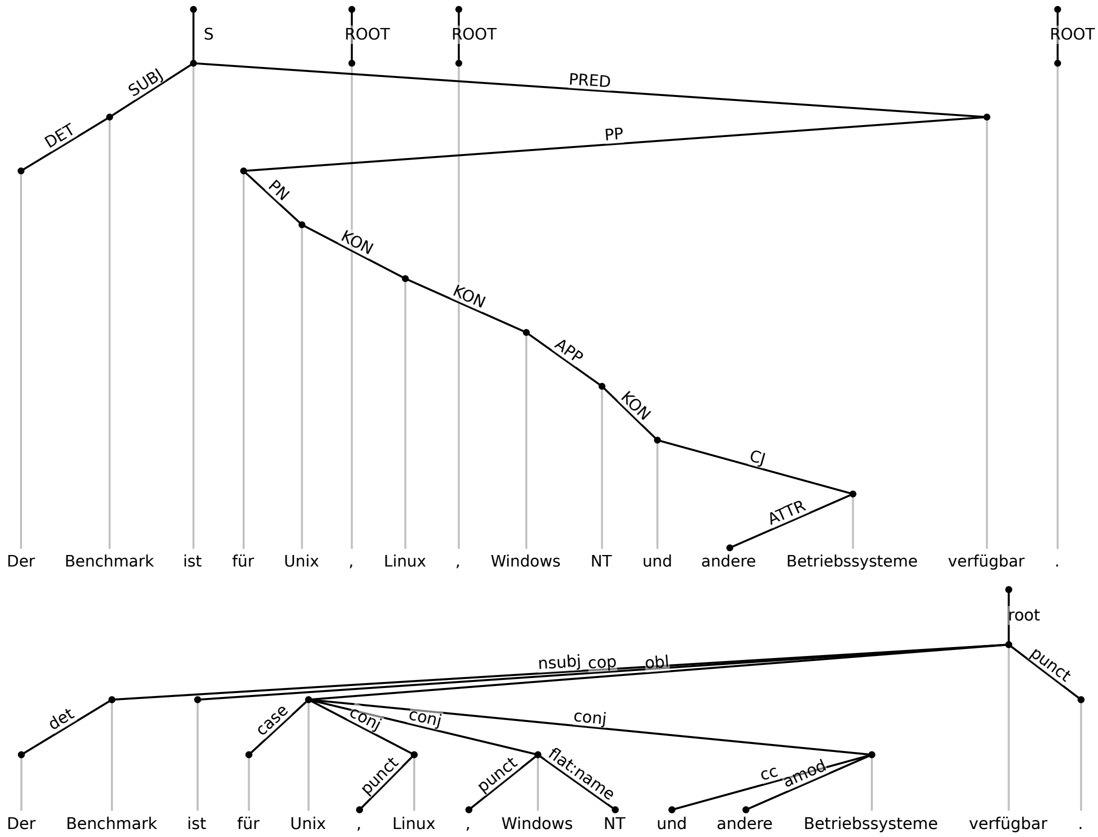

S
ROOT
ROOT
ROOT
SUBJ
DET
PRED
PP
PN
KON
KON
APP
KON
CJ
ATTR
Der Benchmark ist für Unix , Linux , Windows NT und andere Betriebssysteme verfügbar .
root
nsubj
cop
obl
punct
det
case
conj
conj
conj
punct
punct
flat:name
cc
amod
Der Benchmark ist für Unix , Linux , Windows NT und andere Betriebssysteme verfügbar .

Figure 2: A dependency structure in the HDT schema (top) and the corresponding UD tree (bottom).

schemas. In the next sections, we will go into more detail about the difficulties caused by this and how we solved them.

Since Hennig and Köhn (2017) already covered the most common structures, a significant part of the recent conversion effort consisted in creating rules for edge cases and exceptions to the preexisting general rules as well as augmenting the preexisting rules using Groovy code in cases where they needed more complex predicates. We implemented a total of 99 conversion rules for dependency conversion. 34 rules are simple one-to-one mappings converting a single node considering only the previous relation and PoS tag. In about one quarter of the rules we make use of Groovy code to condition matching in ways not expressible otherwise. Usually this was only necessary to add trivial predicates, applying the tree transducer formalism as usual. The only exception to this are coordinating conjunctions, which we will go over in more detail after commenting on some of the simpler extended rules.

4.1 Considering information difference

We mentioned earlier that HDT and UD value different kinds of information. Even though in some cases, like with the 34 simple one-to-one rules, the dependency relations are equivalent and can be transduced easily, the rules often do have to make significant adaptations. The difference in focus between the two schemas means that, when converting dependency trees from the HDT to UD, we face three possible structural complications: incongruence when equivalent structures are represented differently across the two annotation schemas, information deficit when information not contained in the HDT is required by UD and information loss when information contained in the HDT is not kept in UD.

Incongruence can usually be solved by standard TrUDucer rules since the information contained in both representations is the same. The dependency trees simply have to be adapted as described in the Section 3; by changing labels, attaching relations in different places, inverting them and so on.

Information deficit often requires rules with the additional predicates Groovy code can provide because the information needed for the UD relations is not contained in the corresponding HDT relations. Still, the information needed can be extracted from the relations' immediate context most of the time. In one case this was not enough, so we used Wiktionary with minimal manual corrections to supplement our data and determine the correct conversion.

Information loss does not affect the conversion process since it does not impact the well-formedness of the resulting UD relations. It is also only information loss in the sense that the information is not contained it the dependency relations anymore. Much of it is still retained as part of the universal features.

Objects Objects are a good example for both information loss and one-to-one matching. The HDT uses seven different relation labels involving objects, covering phenomena like case (genitive, dative or accusative) and constructions with several objects. UD only differentiates between (direct) objects and – if there are several objects in the same phrase – indirect objects. Genitive, dative and accusative objects are converted to *obj* (or to *iobj* if applicable) and case information is mostly retained in the UD features. The other structures involving objects receive other labels, often depending on context: object clauses are converted to *ccomp*, object infinitives (an infinitive as complement of another verb) to *xcomp* etc.

Copula verbs An example of incongruence and of those simple extended predicates are copula verbs, which are not annotated in the HDT but required by UD - as shown in Figure 2. However, UD only accepts a maximum of one copula for most languages, usually a form of to be (called "pure copula") which is annotated as *cop*.[5] For disambiguation of copula and non-copula verbs, a check of the verb's lemma is required, which is a predicate not included in the TrUDucer rule syntax but easily added by implementing it in the Groovy code.

APP relations The *APP* relation is a notable beneficiary of the expanded predicates that Groovy code offers – converting these relations turned out to be particularly difficult since some PoS tags are unreliable in the HDT. This lack of reliability makes it harder to correctly disambiguate between the target UD relations in these cases since in the HDT, all consecutive constituents of noun phrases (as long as they are not determiners or attributes) are subordinated to their precursor as *APP* (Foth, 2006, p. 13f.), which leads to highly similar structures with the main distinguishing feature being those unreliable PoS tags.

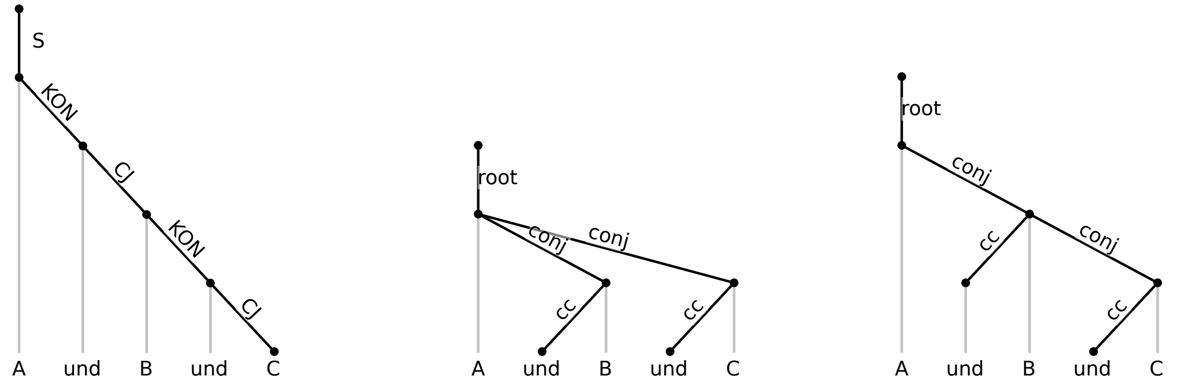
The standard case of the *APP* relation is an apposition, which is annotated as *appos* in UD. However, as can be seen in Figure 2, the *APP* relation is also used for the constituents of a noun phrase in general, for example in names or dates which consist of several tokens and whose UD relation is *flat*. We mainly recognise *flat* relations using certain PoS tags – they will be further discussed in Section 4.4. For *appos*, we use different predicates.

appos is either used when an *APP* relation is interrupted by punctuation, or as a general fallback rule when neither this nor the PoS tags resulting in *flat* apply. Since punctuation is not part of the HDT dependency relations, we used Groovy code to incorporate punctuation into the rules and help with recognising appositions.

Inherently reflexive Verbs Another interesting benefit of embedding Groovy code to extend the predicates is the availability of complex data structures for rule matching. One language feature required in the UD annotation schema but not encoded in the HDT (information deficit) is the special annotation of inherently reflexive verbs – their objects are annotated as *expl:pv* instead of *obj*. The HDT does not annotate reflexivity at all. While finding a reflexive use of a verb is trivial, differentiating between inherently reflexive and non-inherently reflexive verbs requires knowledge about the verb and its use context. This is impossible to achieve by using simple predicates in the treebank conversion because no such knowledge about the verb is given, but we can look through a list of verb usage generated by parsing the German Wiktionary[6] entries for verbs instead. The collaborative nature of Wiktionary means that the data is semi-structured, making information more difficult to extract. Also, verbs often are only inherently reflexive in specific contexts which are not easy to disambiguate automatically. Sometimes, entries in the German Wiktionary are simply incomplete, missing features like reflexivity completely. We manually corrected parts of the list of verb usage to prevent erroneous conversions.

[5] `https://universaldependencies.org/u/dep/cop.html`

[6] `https://de.wiktionary.org/`



(a) HDT conjunction: (A and B and C), ((A and B) and C) or (A and (B and C))

(b) Standard UD conjunction: either (A and B and C) or ((A and B) and C)

(c) Explicitly nested UD conjunction: this is unambiguous and only means (A and (B and C))

Figure 3: Conjunction ambiguities in HDT and UD annotations

4.2 Converting complex structures – coordinating conjunctions

One of the more interesting cases that require complex rules as well as scripting concerns coordinating conjunctions. As seen in Figure 2, the HDT treats coordinating conjunctions as long chains, with each following conjunct being subordinated to the prior. Most other dependency relations tend to branch off, leaving this chain intact, but the aforementioned *APP* relation is also annotated as a single chain of dependency relations in the HDT and regularly interrupts conjunction chains.

In contrast, UD also treats the first conjunct of a conjunction as the head, but requires that "all the other conjuncts depend on it via the conj relation"[7], creating a structure with several parallel branches instead of one long branch (respectively chain).

Issues with identifying the structures To convert these conjunctions to UD, the individual conjuncts need to be pulled up. This is straightforward enough for pure conjunctions, but there are two major issues.

The first issue is the interruption of conjunction chains by the APP relation (Figure 2). It is difficult and awkward to match and convert these structures using only tree transducers since the interrupting APP chains have no fixed length. Unlike with the simpler cases, we don't only use Groovy code here to expand rule predicates, but also for the rule itself. Groovy code allows us to check whether any child nodes of a conjunction interrupted by any number of APP relations are also conjunctions and, if they are, recognise them as part of the original conjunction and pull them up. Sadly, this is not a perfect solution because of the possibility of nested coordination, which leads us to our second issue: ambiguity.

When there are multiple conjuncts linked by conjunctions (as opposed to being linked by punctuation, as is commonly the case for longer conjunctions), there tends to be some ambiguity concerning their exact hierarchy. Figure 3 shows how this ambiguity is represented in HDT and UD with the general example phrase "A and B and C" and its three possible meanings: meaning 1 (A and B and C), meaning 2 ((A and B) and C) and meaning 3 (A and (B and C)).

Figure 3a shows how the example phrase would be annotated in the HDT. It would be thinkable to distinguish meaning 1 form meanings 2 and 3 by, for example, linking "B" to its previous conjunction via the *KON* relation instead, allowing the example phrase to have the same dependency structure as the phrase "A, B and C" if deemed more appropriate by the annotator. But since the HDT guidelines require a conjunct following a conjunction to always be attached to the conjunction via the *CJ* relation, the HDT is completely unable to distinguish between the phrase's three possible meanings (Figure 3a).

The UD representation is slightly less ambiguous. Meanings 1 and 2 are still annotated the same way (Figure 3b), but meaning 3 is represented by a different structure than the other two (Figure 3c).

Conversion details When converting such structures, we face the problem that the given dependency relations contain less information than the target dependency relations require. It is not impossible to

[7] `https://universaldependencies.org/u/dep/conj.html`

disambiguate such cases, but this generally requires access to, and understanding of, semantic information and context. In practice it would be necessary to check every conjunction manually to always correctly disambiguate them. For now, we settled on an automatic conversion without user input that converts as many conjunctions correctly as possible. Meaning 1 is by far the most common one in the HDT. Conjunction words generally are replaced by punctuation in longer conjunctions and in this case they are correctly converted to the corresponding UD structure shown in Figure 3b. The cases where we have meaning 1 coincide with the dependency structure of Figure 3a, and which would lead to a wrong conversion, are quite rare to nonexistent. This leaves meanings 2 and 3. Meaning 2 should be converted to structure 3b, but will by default be converted to structure 3c. Meaning 3 should and will be converted to the structure shown in Figure 3c.

In conclusion, the interruptions of *KON* relations by *APP* relations make their conversion more tricky, but pose a solvable problem. The ambiguity inherent to nested coordination remains an issue.

4.3 Decisions based on user input

For some rules we can not decide whether they should be applied even by looking at the global sentence context; they need user input. We re-use the ability to evaluate Groovy predicates by having functions that yield to the user and use their response. The answer is stored in a database so that the user does not need to be asked again on subsequent conversions. For example, the rules need to yield to the user when converting inherently reflexive verbs which are not in the dictionary and therefore cannot be told apart automatically. We are also considerung using rules that yield to user input for prepositional phrases, where in some cases the PoS tag is not sufficient to disambiguate the dependants below nominals and the dependants below predicates. Rules with interactive user input requests are implemented for both mentioned cases but are not used in the current conversion. The amount of manual labour needed even for these edge cases would still be substantial because of the size of the Hamburg Dependency Treebank. Right now the edge cases will fall back to default conversions until we either find a way to automatically distinguish these cases or we manually decide the relation for each occurrence once.

Fragmented sentences Due to the nature of fragments in the HDT, a default fallback rule is not an option in those cases. The HDT uses the label S for the root of a tree as well as the root of a fragment (e.g. a parenthesis, or more general a construction that cannot be integrated into the tree structure without breaking hard constraints imposed by the HDT annotation schema). In the HDT, it is acceptable for sentences containing these structures to have multiple roots, but UD requires sentences to have a single root. Our current conversion process is able to recognise these structures and convert the individual subtrees correctly, but is not yet able to automatically attach the fragments correctly due to the sheer number of possibilities.

Punctuation Another use case of the Groovy scripting language embedded in the rule file is the annotation of punctuation in the sentence. Punctuation is not annotated in the original HDT and therefore can not be converted by the TrUDucer rules as it is not part of the well-formed tree required by the TrUDucer rules. However, annotation of punctuation in universal dependencies is defined by four precise rules[8]. With heuristics we were able to apply those rules to the otherwise converted treebank. In some German sentences punctuation can be ambiguous. Whenever the heuristics discover such ambiguities, they will not annotate them. Punctuation for which no heuristics apply will also not be annotated. Example of that are unbalanced quotation marks or parentheses. Punctuation in these sentences will have to be manually annotated; until then, they are excluded from the release of the converted treebank.

4.4 Features

While the main focus of the TrUDucer software is converting dependency trees, which PoS tag and morphology conversion is not a part of, the software can also apply given lookup tables for conversion of said features. They are applied in an additional step after the dependency conversion. The annotation of morphology in HDT and UD is quite similar, therefore we can convert many features with another

[8] https://universaldependencies.org/u/dep/punct.html

one-to-one conversion schema. For now, features not directly encoded in the HDT are not annotated in the UD conversion. The tree transducer rules often rely on morphological information, and while morphology was not the focus of the HDT's original annotation, it still contains a lot of information which is lost or more difficult to access after a complete conversion due to the UD annotation being more coarse-grained than the HDT one. Thus we convert the morphology (in particular the PoS tags) after the dependency relations.

Part of Speech tags For conversion of Part of Speech tags, we use a simple mapping. As the HDT uses the Stuttgart-Tübingen-Tagset (Schiller et al., 1999) for PoS tags, we adapted the lookup table used in Çöltekin et al. (2017), who converted the TÜBA-D/Z, which uses the same tagset. An exception for the simple lookup is the PoS tag *PIDAT* (attributive indefinite pronoun with a determiner)[9] which, depending on the dependency relation, should either be converted to *DET* or to *ADV*.

Unfortunately, there are some problems with the source annotation. PoS tags and morphology were not a focus of the original annotation effort and therefore have reduced quality. Particularly unreliable are the annotations of foreign noun phrases. The three main PoS tags for nouns in the HDT are NN (normal noun), NE (proper name) and FM (foreign material). The HDT documentation states that the distinction between them is often difficult - when in doubt, the tokenizer was supposed to classify a noun phrase's tokens the way it would have if they were isolated (Foth, 2006, p. 43f.), meaning that their PoS tags differ on a case by case basis. On top of making it harder to distinguish between *appos* and *flat* when converting the *APP* relation, as mentioned in section 4.1, it also complicates the correct use of the *flat* relation's subtypes: *flat:name* and *flat:foreign*. The particular lack of reliability of PoS tags involving foreign words makes it impossible to systematically recognize foreign phrases as such and select the correct sub-type for them. Thus, we have not yet implemented *flat:foreign*. *flat:name* is somewhat less error prone.

5 Statistics and Evaluation

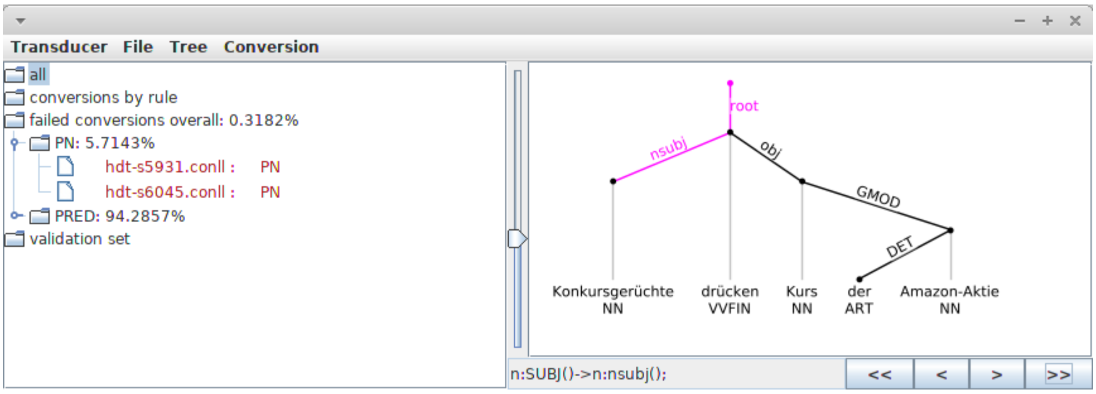
We manually annotated a set of 50 sentences held out from the rule generation process to evaluate the quality of the converter. The sentences were chosen by randomly sampling the part B of the HDT and also used as validation in Hennig and Köhn (2017). Our manual annotations differ slightly because of changes to the current UD standards. Punctuation marks (84 out of the 782 tokens) were ignored during this comparison. Looking at the 698 remaining converted words, 558 matched the hand-annotated dependency relations, leaving 127 words not matching the gold standard. Further analysis showed that the dependency label was actually correct in 88 of these cases but the transducer gave the dependency relation an additional (correct) subtype which was not given in the set of hand-annotated sentences. Some of the hand-annotated relations turned out to be wrong when comparing them to the result from the transducer. In the end, neglecting the punctuation marks, for 679 out of 698 words the automatically converted annotations matched the corrected hand-annotations, yielding a labeled accuracy of 97.3%. We further evaluated the conversion as performed by Seddah et al. (2018): by critically looking through 100 randomly selected sentences and checking for annotation and conversion errors. The resulting accuracy confirms the previous evaluation. Overall, 71% of the evaluated sentences where converted without any errors and 1506 of the 1548 non-punctuation dependencies where converted correctly, again yielding an accuracy of 97.3%.

This accuracy is significantly higher than other reported conversion accuracies; Seddah et al. (2018) e. g. report a labeled conversion accuracy of 94.75% and 93.27% on their held-out sets, which is twice the amount of labeled errors.

6 Interactive Workbench

In the context of dependency trees, the aforementioned transducer rules are applied to convert dependency relations and change the dependency heads. While the TrUDucer software itself is treebank-agnostic, the conversion rules are conversion specific. The development of those treebank-specific rules takes a big proportion of the effort put into converting the treebank. Therefore it is a huge advantage to have the

[9] `https://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html`



all
conversions by rule
failed conversions overall: 0.3182%
PN: 5.7143%
hdt-s5931.conll : PN
hdt-s6045.conll : PN
PRED: 94.2857%
validation set

```
n:SUBJ()->n:nsubj();
```

Figure 4: The TrUDucer GUI. Left: a selection of different overviews. All sentences of the treebank, Conversion steps sorted by rules applied, trees which could not be completely converted (sorted by the label of the first edge that could not be converted), all sentences of the validation set (comparison against a gold standard). Right: Interactive visualization of a selected conversion. Edges converted in this step are highlighted, unconverted edges with all-caps dependency labels, the rule applied in the current step is shown at the bottom.

process of developing conversion rules be as effortless as possible, which we tried to achieve with the graphical interface, the interactive workbench, for treebank conversion analysis.

One of the benefits of the interactive workbench is the support for treebank search queries. During the development of conversion rules, tests and validations are of huge importance to assess whether all rules work as intended and which structures are not yet converted correctly. Therefore, we added a graphical interface to TrUDucer that allows to search through the converted and unconverted treebank, highlight conversion problems and give additional insights on each individual conversion rule. For each conversion, it can visualize the rule applications and for each rule it shows exactly where it was applied. The latter turned out to be the most important information to assess the effect of each new rule and its interaction with already existing rules with minimal effort.

To further facilitate introspection, TrUDucer implements a filter over all the sentences of the treebank for the applied rules, either by searching for a specific phrase or by searching for subtrees in the dependency structure. The latter is able to search through both the converted and unconverted treebank. This was an important aid in checking the correctness for specific and possibly rare grammatical structures which were otherwise hard to find.

In addition to manually checking new rules by filtering the treebank for structures and applied rules, a regression check using sentences with manually created gold standard is used. Whenever a conversion is considered correct by an annotator, they can add that sentence to the gold standard. The regression check performs the conversion of the source annotation again and checks it against the gold standard. This process is usually run after the rule set was changed to check whether the modifications introduce unwanted side effects. This check is part of the GUI and allows to visually compare the gold standard annotation with the automatically converted annotation, highlight differences and modify and extend the gold standard within that interface, if needed.

The same graphical comparison of two dependency trees turned out to also be usable to visually represent a conversion rule, as the matching pattern is also given in form of a tree and can be compared to the replacement pattern as shown in Figure 5. By visualizing the tree-structure of both sides of a rule, we were able to find mistakes in newly written rules as it helps to get a more natural understanding of each rule.

TrUDucer can be used with the GUI for interactive rule development and in batch-mode to convert a whole treebank. As the treebank-specific rules are decoupled from the software itself, we hope to have created a software which is usable for flexible conversion of different dependency treebanks instead of just the HDT.

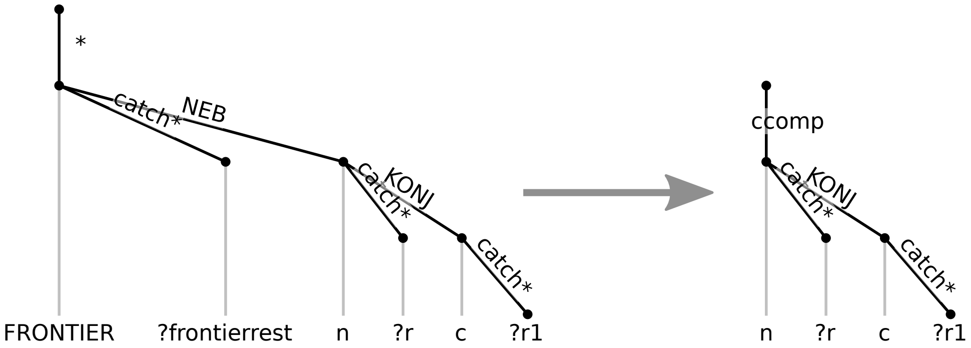


Figure 5: The transducer rule "n:NEB(c:KONJ()) -> n:ccomp(c())" visualised. Noticeably it also shows the frontier node in the matching tree and the catchall nodes (edges labeled with catch*) that are implicitly defined in the rule to allow better understanding of the interactions of the rule. Catchall nodes are special nodes in the rule tree in that they can match with multiple conversion tree nodes simultaneously.

7 Conclusion and Outlook

While we can convert 99.7% of the sentences during dependency conversion, only 90,7% of the treebank have been released at the time of publication. The large difference between dependency conversion coverage and overall conversion coverage is due to our focus being mainly towards dependency conversion, with morphological features and PoS tags only treated afterwards (as mentioned in Section 4.4). We only included sentences passing the UD validator, which in most cases of rejection complained about unattached fragments (in about 2% of all converted sentences), punctuation (estimated 3%) as well as incongruities between PoS and dependency relations.

For a complete conversion of the HDT to UD, there are still a number of things that need to be done. We need to find a way to (at least semi-) automatically resolve dependence ambiguity. This includes attaching fragments. The necessary infrastructure for manual attachment is already in place. However, an at least semi-automatic conversion would be preferable – if possible – due to the large number of fragments and their diversity. 0.9% of the sentences in the treebank contain at least one fragment, not considering fragmented punctuation marks. Similarly, we need to resolve the remaining issues with the conversion of *KON* and *APP* relations: we need to assess where the resulting relations are attached and make sure that the fallback rules minimise wrong attachments. We also need to improve PoS tag quality concerning noun types. We currently plan on doing this by training a tagger on more accurate data and using it to correct the noun-related PoS tags in the HDT, using the active learning approach proposed by Rehbein and Ruppenhofer (2017).

The rules for exceptions and difficult fringe cases concerning prepositional phrases need to be completed. This concerns cases where it is unclear whether the prepositional phrase should receive the *nmod* or *obl* label. Currently, we use *obl* for prepositional phrases attached to a predicate and *nmod* as a general fallback rule since the regent tends to be a nominal in the remaining cases. We plan to add specific rules for regents with other PoS tags as well to increase accuracy. These rules will most likely ask for user input since these cases are often difficult even for a human annotator due to their high ambiguity.

Composite pronouns need to be split into syntactic words[10], and fixed multi-word expressions need to be implemented. They are represented by the *fixed* relation in UD, but not annotated in the HDT. In an ongoing conversion of the TIGER corpus (also using TrUDucer), they are implemented using a list (Watter, 2018). We will probably use a similar solution.

We hope that our conversion of the HDT further establishes German as a UD language and plan to expand the German UD documentation by contributing our findings.

Acknowledgements We would like to thank the three anonymous reviewers for their helpful remarks.

[10] https://universaldependencies.org/u/overview/tokenization.html

References

Lars Ahrenberg. 2015. Converting an English-Swedish Parallel Treebank to Universal Dependencies. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 10–19, Uppsala, Sweden, August. Uppsala University, Uppsala, Sweden.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2(4):597–620.

Çağrı Çöltekin, Ben Campbell, Erhard Hinrichs, and Heike Telljohann. 2017. Converting the TüBa-D/Z Treebank of German to Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 27–37, Gothenburg, Sweden, May. Association for Computational Linguistics.

Kilian A. Foth, Arne Köhn, Niels Beuck, and Wolfgang Menzel. 2014. Because Size Does Matter: The Hamburg Dependency Treebank. 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 Language Resources and Evaluation Conference 2014*, pages 2326–2333, Reykjavik, Iceland, may. LREC, European Language Resources Association (ELRA).

Kilian A. Foth, 2006. *Eine umfassende Constraint-Dependenz-Grammatik des Deutschen*.

Felix Hennig and Arne Köhn. 2017. Dependency Tree Transformation with Tree Transducers. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 58–66, Gothenburg, Sweden, May. Association for Computational Linguistics.

Anders Johannsen, Héctor Martínez Alonso, and Barbara Plank. 2015. Universal dependencies for danish. In Markus Dickinson, Erhard Hinrichs, Agnieszka Patejuk, and Adam Przepiórkowski, editors, *Proceedings of the Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT14)*, pages 157–167, Warsaw, Poland, December.

Andreas Maletti. 2010. Survey: Tree transducers in machine translation. In Henning Bordihn, Rudolf Freund, Thomas Hinze, Markus Holzer, Martin Kutrib, and Friedrich Otto, editors, *Proc. 2nd Int. Workshop Non-Classical Models of Automata and Applications*, volume 263 of `books@ocg.at`, pages 11–32. Österreichische Computer Gesellschaft.

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. 2013. 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. Association for Computational Linguistics.

Joakim Nivre. 2014. Universal dependencies for swedish. In *Proceedings of the Swedish Language Technology Conference (SLTC)*, Uppsala, Sweden, November. Uppsala University, Uppsala, Sweden.

Lilja Øvrelid and Petter Hohle. 2016. Universal Dependencies for Norwegian. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.* European Language Resources Association (ELRA).

Sampo Pyysalo, Jenna Kanerva, Anna Missilä, Veronika Laippala, and Filip Ginter. 2015. Universal Dependencies for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 163–172, Vilnius, Lithuania, May. Linköping University Electronic Press, Sweden.

Ines Rehbein and Josef Ruppenhofer. 2017. Detecting annotation noise in automatically labelled data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1160–1170, Vancouver, Canada, July. Association for Computational Linguistics.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Universität Stuttgart / Universität Tübingen.

Djamé Seddah, Eric De La Clergerie, Benoît Sagot, Héctor Martínez Alonso, and Marie Candito. 2018. Cheating a Parser to Death: Data-driven Cross-Treebank Annotation Transfer. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan, May. European Language Resource Association.

Umut Sulubacak, Memduh Gokirmak, Francis Tyers, Çağrı Çöltekin, Joakim Nivre, and Gülşen Eryiğit. 2016. Universal Dependencies for Turkish. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3444–3454, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Juhi Tandon, Himani Chaudhry, Riyaz Ahmad Bhat, and Dipti Misra Sharma. 2016. Conversion from Paninian Karakas to Universal Dependencies for Hindi Dependency Treebank. In Katrin Tomanek and Annemarie Friedrich, editors, *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016, LAW@ACL 2016, August 11, 2016, Berlin, Germany*. The Association for Computer Linguistics.

Heike Telljohann, Erhard Hinrichs, and Sandra Kübler. 2004. The Tüba-D/Z treebank: Annotating German with a context-free backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 2229–2235.

Francis M. Tyers and Mariya Sheyanova. 2017. Annotation schemes in North Sámi dependency parsing. In *Proceedings of the Third Workshop on Computational Linguistics for Uralic Languages*, pages 66–75, St. Petersburg, Russia, January. Association for Computational Linguistics.

Camille Watter. 2018. Converting the german constituency TIGER treebank to the universal dependencies format. Bachelor's thesis, Universität Zürich.

Nested coordination in Universal Dependencies

Adam Przepiórkowski[1,2,3] **Agnieszka Patejuk**[1,4]
[1]Institute of Computer Science, Polish Academy of Sciences
[2]Institute of Philosophy, University of Warsaw
[3]Wolfson College, University of Oxford
[4]Faculty of Linguistics, Philology and Phonetics, University of Oxford
adamp@ipipan.waw.pl aep@ipipan.waw.pl

Abstract

The aim of this paper is to extend the representation of coordination in Universal Dependencies in a way that makes it possible to distinguish between different embeddings in coordinate structures.

1 Introduction

One principled problem with the UD approach to coordination known to the Universal Dependencies (UD; http://universaldependencies.org/; Nivre et al., 2016) community concerns nested – i.e. immediately embedded – coordination, as in:[1]

(1) Tom and Jerry and Scooby-Doo

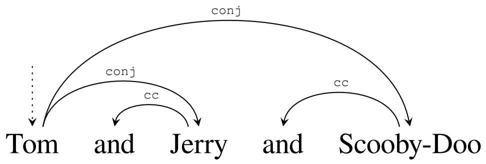
There are three possible ways to structure (1):

(2) Tom and Jerry and Scooby-Doo
(3) [Tom and Jerry] and Scooby-Doo
(4) Tom and [Jerry and Scooby-Doo]

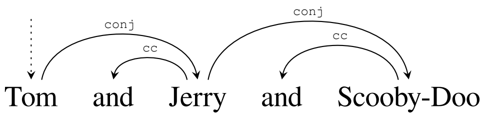
That is, (1) may be construed as flat ternary coordination (cf. (2)), or as binary coordination whose first (cf. (3)) or second (cf. (4)) conjunct is itself a coordinate structure.

UD is not able to distinguish the first two structures, (2)–(3) – it assigns the same representation (5) to both, while representing (4) as (6):

(5)



(6)



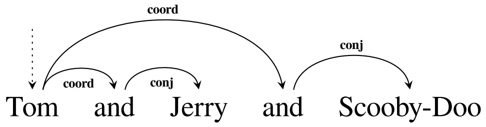
Przepiórkowski and Patejuk, 2019, §4.3 observe that this is not a fundamental problem in practice: in about a dozen cases of nested coordination (out of over 17,000 sentences) in the UD_Polish-LFG treebank of Polish, all or almost all involve a contrastive conjunction such as *ale* 'but', which is strictly binary. Hence, it is often easy to 'disambiguate' representations such as (5) to a truly nested structure: if one of the conjuncts is binary, this cannot be a flat ternary structure. Perhaps for this reason this theoretical flaw has been tolerated in the UD community. Nevertheless, as argued e.g. in Borsley, 2005, 468–469, sequences such as (1) are truly ambiguous between the structures indicated in (2)–(4), so it would be at least theoretically desirable for UD to be able to represent such different nestings.

2 Nested Coordination in Dependency Grammars

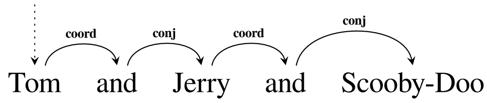
A similar problem occurs in the surface syntactic dependency representation of coordination assumed by Igor Mel'čuk's Meaning–Text Theory (MTT), where conjuncts and conjunctions form a chain; on that approach, the structure (3) is distinguished from the other two, as it gets the basic representation in (7), but the other two structures, (2) and (4), share the basic representation in (8):

[1] See http://universaldependencies.org/u/dep/conj.html#nested-coordination (last referenced on 16 June 2019), where this problem is explicitly pointed out.

(7)



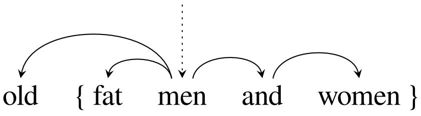
(8)



Contemporary dependency theories deal with various coordination problems by giving up pure dependency representations and introducing additional constituency-like structures: *groupings* in MTT[2] and *word strings* in Richard Hudson's Word Grammar (WG) and related dependency approaches.[3] Groupings are unordered sets containing nodes of contiguous dependency trees. They seem to be used in MTT if and only if the need arises to indicate that a coordinate structure should be treated as a whole. Mel'čuk (2009, 100) illustrates this need with example (9) on the reading where both men and women are described as old, but only men as fat. On this reading, the representation of (9) is (10) (ignoring dependency labels):

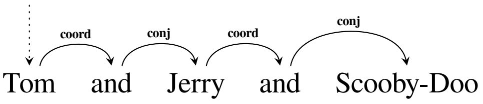
(9) old fat men and women

(10)

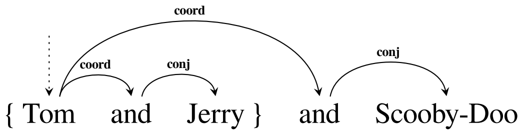


On this view, groupings are superimposed on a well-formed dependency tree, i.e. groupings do not act as nodes in dependency trees. When applied to nested coordination, the three representations of (2)–(4) could be (11)–(13), respectively (Mel'čuk, 2009, 101):

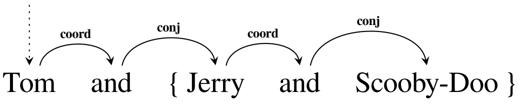
(11)



(12)

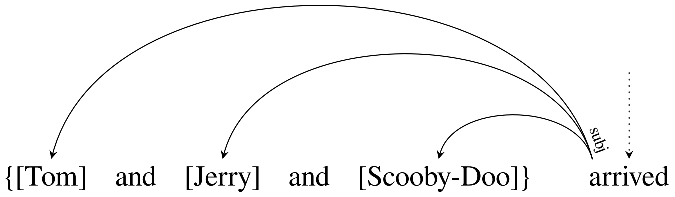


(13)



In Word Grammar, whole coordinate structures and particular conjuncts are analysed as constituents, called 'word strings': they may contain words or other 'word strings', i.e., they may actually have a hierarchical structure. Such 'word strings' are marked with curly brackets in the case of coordinate structures and by square brackets in the case of those conjuncts which are not coordinate structures themselves:

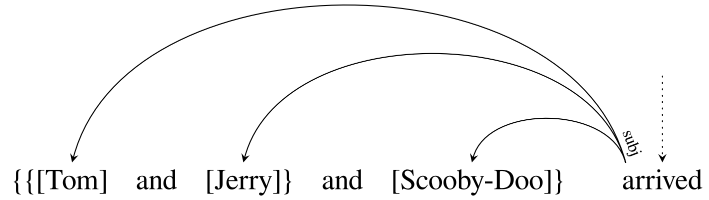
(14)



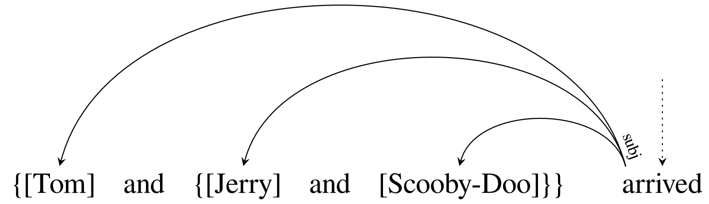
[2] See e.g. Mel'čuk, 1964, 25, Mel'čuk, 1974, 214–216, Mel'čuk and Pertsov, 1987, 74, Mel'čuk, 1988, 28–33, Mel'čuk, 2009, 94, 100–101.

[3] See e.g. Hudson, 1980, 496–499, Hudson, 1984, 211–240, Hudson, 1988, 1989, Hudson, 1990, 404–421 and Hudson, 2018, §4.2, as well as Pickering and Barry, 1993 and Osborne, 2006b,a, Osborne and Groß, 2017. For a more general combination of dependency and constituency relations, with applications to coordination, see Kahane, 1997 and references therein to 1960s work by Aleksej V. Gladkij.

(15)



(16)



The use of constituency to differentiate between different nestings of coordinate structures is suggested e.g. in Hudson, 1988, 318 and in Hudson, 1990, 408. On this approach, coordinate structures are not connected dependency graphs. The status of conjunctions differs in different versions of WG, but in Hudson, 1984, 1988, 1989, 1990 they are not integrated with the rest of the *dependency* structure: they are neither heads nor dependents. On the other hand, conjunctions are integrated in the *constituency* structure of coordinate structures: they are immediate constituents of coordinate structures (in Hudson, 1984, 1988, 1989 and in (14)–(16)) or of the immediately following conjuncts (in Hudson, 1990). Apart from constituency structures used to model coordination, another formal device which goes beyond simple dependency structures is that of split dependency relations. Thus, all three structures (14)–(16) should be understood as containing a single subject dependency originating in *arrived*, which splits into three edges targeting the proper nouns.

3 Nested Coordination in Enhanced UD

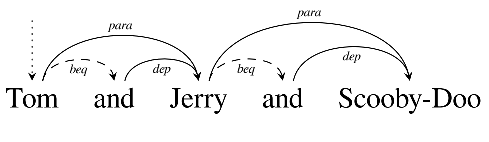
UD does not assume either constituency structure or split relations; basic UD representations are simple dependency trees, and enhanced representations are dependency graphs (with the option of introducing empty nodes). How could then the three different structures of (1) be represented in UD? In the following subsections, we consider various solutions starting from the least UD-conservative (i.e., least conservative from the point of view of UD) and moving to the most UD-conservative.

3.1 Different Topology

A theoretically possible solution would be to change the general UD topology of coordinate structures and represent them as headed by the conjunction. While this was probably the most popular representation of coordination in pre-UD treebanks (Popel et al., 2013), the idea that conjunctions head coordinate structures is widely rejected on theoretical linguistic grounds, both within dependency approaches (e.g. Mel'čuk and Pertsov, 1987, 65, Hudson, 1988, 314–315 and Gerdes and Kahane, 2015, 102–105) and within constituency approaches (Borsley, 2005). Hence, we will not consider this possibility here.

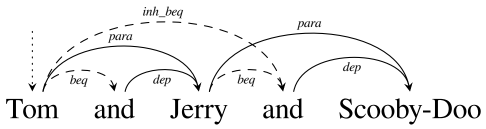
A proposal to distinguish different nestings in dependency graphs which does not assume mechanisms outside of dependency relations and in which coordination is represented as headed by the first conjunct is outlined in Gerdes and Kahane, 2015, 108. According to that proposal, the three representations of different nestings of *Tom and Jerry and Scooby-Doo* would be:[4]

(17)

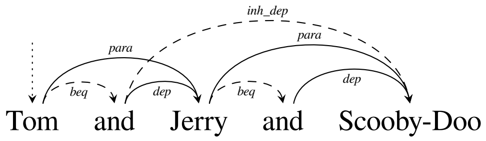


[4] See Gerdes and Kahane, 2015 for the explanation of the labels occurring in these graphs: *para(digmatic link)*, *dep(endent)*, *beq(ueather)*, *inh(erited)_dep(endent)* and *inh(erited)_beq(ueather)*; their precise meaning is not crucial here.

(18)



(19)

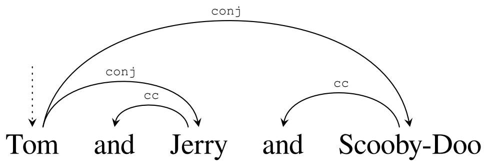


In the above graphs, primary dependencies are shown as solid lines and secondary dependencies are shown as dashed arrows. Note that, even after removing the secondary dependencies, these graphs do not become trees: all non-initial conjuncts have two incoming edges. Moreover, reversing the *dep* dependencies between conjunctions and the following conjuncts does not necessarily produce UD-like trees. The reason is that the "chain" ("Moscow", in the terminology of Popel et al., 2013) topology of (17) (representing no nesting), in which conjuncts form a dependency chain, differs from the "bouquet" ("Stanford") topology adopted in UD, in which all non-initial conjuncts dependent on the first one; and similarly for (18) (representing nesting indicated in (3)). While Gerdes and Kahane (2015) carefully justify such a representation of coordinate (and, more generally, paradigmatic) structures, the solutions presented below are more UD-conservative.

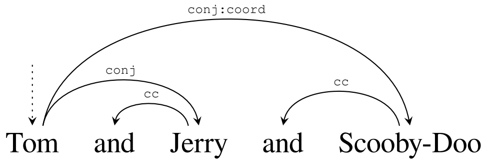
3.2 Enriching Dependency Labels

One possible solution (suggested to us by Nathan Schneider, p.c., August 2018) is to enrich dependency labels via subtyping, e.g.:

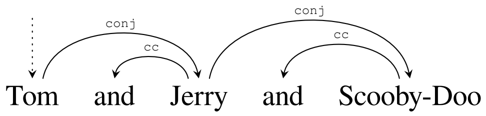
(20)



(21)



(22)

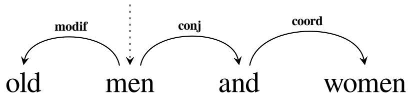


According to this idea, the flat structure is represented as before (see (20), same as (5) above), and so is the structure where the second and third nouns form a nested coordinate structure (see (22), same as (6) above). However, the structure with the first two nouns forming a nested coordinate structure is distinguished from the flat coordination with the use of the subtyped dependency relation `conj:coord`, which signals that the head of the dependency is itself a coordinate structure (see (21)).

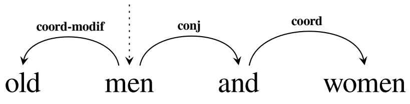
A similar solution to a related problem of distinguishing between the two readings of (23) indicated in (24)–(25) is discussed in Mel'čuk, 2009, 93–94 (and earlier, on the basis of different examples, in Mel'čuk, 1988, 99); the dependency trees corresponding to (24)–(25) are (26)–(27), respectively.

(23) old men and women
(24) [old men] and women
(25) old [men and women]

(26)



(27)



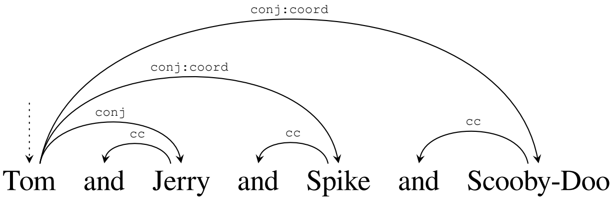
This possibility is rejected as ‘highly unnatural’ (Mel’čuk, 1988, 30) and as leading to the doubling of dependency labels (Mel’čuk, 2009, 94).[5] In fact, when applied to the problem of nested coordination, the `conj` label would not only have to be doubled by `conj:coord`, but – in order to represent more nested coordination – multiplied indefinitely, insofar as there are no theoretical bounds on the depths of nesting of coordinate structures. To see the problem, consider (28), two of its structures indicated in (29)–(30), and the representation in (31):

(28) Tom and Jerry and Spike and Scooby-Doo

(29) [Tom and Jerry] and Spike and Scooby-Doo

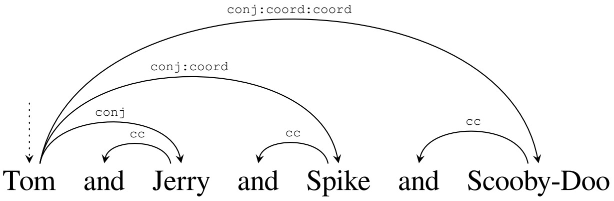
(30) [[Tom and Jerry] and Spike] and Scooby-Doo

(31)



While (31) is a reasonable representation of (29), it would also need to serve as the representation of (30), if `conj` were only allowed to be extended to `conj:coord`. The problem is that `conj:coord` represents the information that the head is embedded inside another coordination, but not the information about the number of coordinate structure boundaries that the dependency crosses. Such information could be represented by repeating the `:coord` subtype an appropriate number of times, as in (32):

(32)

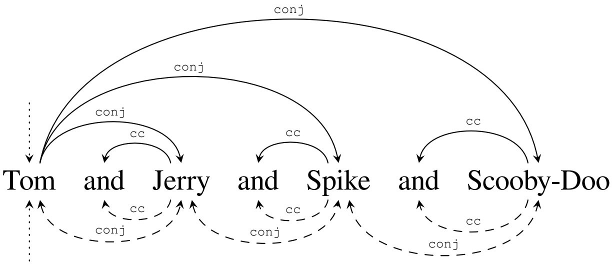


Since adopting this solution would amount to allowing for a theoretically infinite number of possible dependency labels, we reject it here, and instead present two solutions that are free from this problem.[6]

3.3 Co-Headedness of Conjuncts

Another relatively UD-conservative solution is to retain the by now standard basic tree representation of coordination in UD, and distinguish different nesting possibilities in the enhanced representation, by linking co-conjuncts in a bidirectional chain. For example, in the case of *Tom and Jerry and Spike and Scooby-Doo*, the flat structure and the two nestings indicated in (29)–(30) (repeated below) receive the following three UD representations on our proposal, with secondary edges in enhanced representations shown as dashed arrows drawn under the tokens:

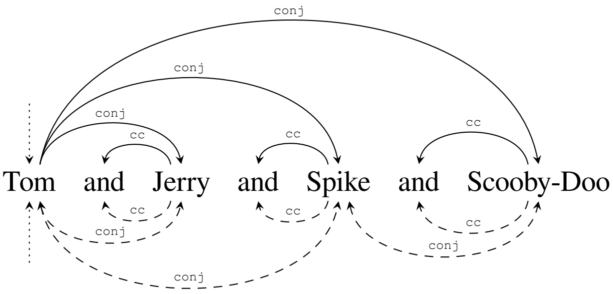
(33) Tom and Jerry and Spike and Scooby-Doo (= (28), i.e., no nesting):



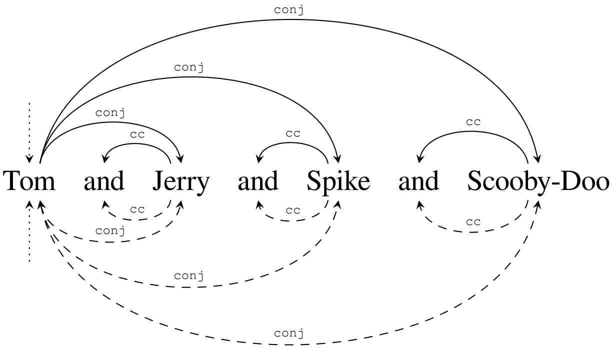
[5] Mel’čuk, 2009, 94 also claims that this solution is ‘not sufficient formally’ to represent the difference between ‘hungry [men and women and children]’ and ‘[hungry [men and women]] and children’, but this difference can in fact be represented by the combination of Mel’čuk’s suggestion concerning the scope of modifiers illustrated in (26)–(27) and its extension to the case of nested coordination discussed here.

[6] Having a limited number of dependency labels is often perceived as important by the UD community; see, e.g., Schuster et al., 2017, 130–131 for arguments against encoding paths in dependency labels in the context of the UD representation of gapping, the most important of which is that this would introduce an unbounded number of dependency relations.

(34) [Tom and Jerry] and Spike and Scooby-Doo (= (29)):



(35) [[Tom and Jerry] and Spike] and Scooby-Doo (= (30)):



According to this proposal, any two neighbouring conjuncts in the same coordinate structure are connected with a bidirectional `conj` dependency in the enhanced representation. In order to avoid introducing a new mechanism into UD, this bidirectional dependency may be understood as two usual (unidirectional) dependencies going in the opposite direction.

The practical advantage of this proposal is that, for any number of conjuncts in a coordinate structure, any two different nestings will provably differ either in their basic tree representation, or in their enhanced representation, or in both. Hence, different nestings of a coordinate structure may now be distinguished in UD; Table 1 gives all 11 possibilities for the case of 4 conjuncts.[7] This would also be true if we did not insist on the bidirectionality, but instead allowed for chaining, say, from left to right. Moreover, this solution is also compatible with the proposal of Kanayama et al. (2018), who convincingly argue for right-headed basic tree representations of coordination in the case of head-final languages such as Japanese and Korean, i.e., representations symmetric with respect to the strictly left-headed trees currently imposed by UD. In the case of such head-final languages, the unidirectional version of the enhanced representation proposed here would make more sense with chains from right to left.

The theoretical advantage which does, however, rely on bidirectionality is related to the frequently expressed (but rarely implemented) sentiment that all conjuncts are heads of a coordinate structure. This is the hallmark of the Generalized Phrase Structure Grammar (GPSG; Gazdar et al., 1985) analysis of coordination (see e.g. Gazdar et al., 1985, ch.6 and Sag et al., 1985), and it has also been defended within Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1987, 1994), e.g. in Abeillé, 2003. This is also a recurrent theme within dependency (and related) approaches ever since their inception: Tesnière, 1959, 2015 (chapters 136 and 143–146) contains analyses of coordinate structures as effectively multi-headed by (roots of) all conjuncts.[8] Similarly, formalisms combining dependency and constituency are sometimes motivated by the need to represent conjuncts as co-heads, cf. e.g. Kahane, 1997, §5.1 and Kahane and

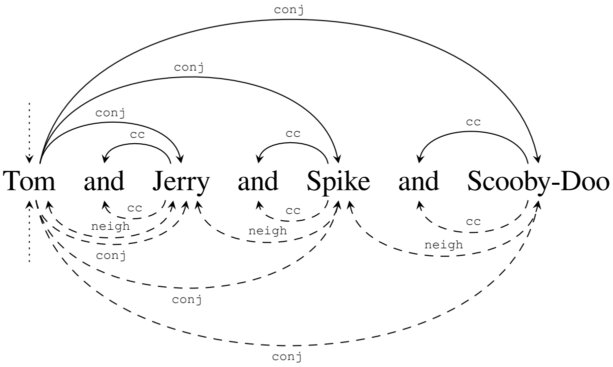
[7] The numbers of different nestings – i.e. 1 (for two conjuncts), 3 (for three conjuncts), 11 (for four conjuncts, as in Table 1), 45 (for five conjuncts), etc. – form a sequence known in combinatorics as (little) Schröder numbers, Schröder–Hipparchus numbers or super-Catalan numbers; see e.g. Stanley, 1997 for the history of these numbers, and their other interpretations. This is an exponential sequence; for example, for ten conjuncts, there are 103,049 possible nestings (as calculated already by Hipparchus of Nicaea, c. 190 – c. 120 BC).

[8] Unlike the representation in chapter 38, which suggests a WG-like analysis.

Mazziotta, 2015, 162–163.[9] Moreover, the reason often given by Hudson for the headless representation of coordination is that – according to Hudson – it is not clear which conjunct should be taken to be the head. The representation proposed here treats all conjuncts (or rather, in Hudson's terminology, conjunct-roots) as heads, even in nested coordination: each conjunct is subordinate to all other conjuncts.[10]

It should be noted that this proposal solves the nested coordination problem on the assumption that the two UD levels of representation – the basic tree and the enhanced graph – are considered simultaneously: neither of the two levels distinguishes all nestings by itself. We do not see a way of modifying the *basic* representation alone in a way that distinguishes all nestings and does not create the problem (discussed above) of an indefinite number of dependency labels. However, the obvious way for the *enhanced* representation to distinguish all nestings alone is to make it contain both: the standard UD representation of coordination and the bidirectional links between neighbouring conjuncts. Assuming such bidirectional links are labelled as `neigh` (in order to distinguish them from the standard `conj` dependencies), the resulting representation of the 'no nesting' case would be as in (36):

(36) Tom and Jerry and Spike and Scooby-Doo (no nesting, all information in enhanced representation):

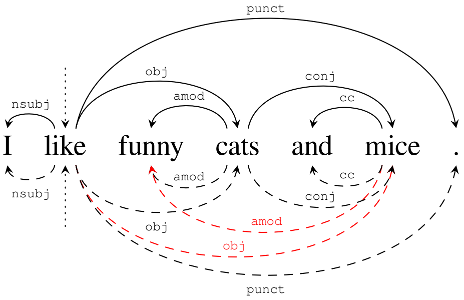


3.4 UD-Conservative Solution

Finally, there is a solution (whose initial version was suggested to us by an anonymous reviewer) that does not have the theoretical advantage of encoding co-headedness of conjuncts, but has the practical advantage of being maximally UD-conservative. As in the previous representation, the basic tree representation of coordination is left intact, but the enhanced representation is a proper extension of this basic tree according to *some* of the general enhanced UD principles of representing coordination. To recall these principles, consider (37) and its UD representation – on the reading where both cats and mice are funny – in (38).

(37) I like funny cats and mice.

(38)



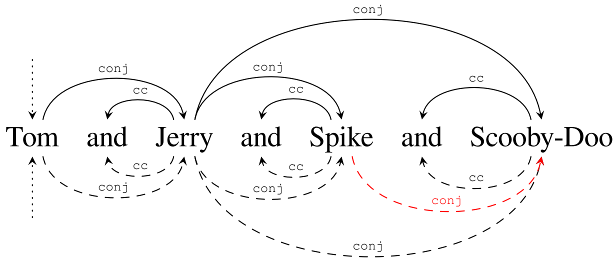
As illustrated in (38), dependencies involving the whole coordinate structure are represented on the head of the coordinate structure in the basic tree, but on all conjuncts in the enhanced graph. This concerns both incoming and outgoing dependencies: the `obj` dependency from the verb targets all conjuncts in the enhanced representation and the `amod` dependency to the adjective originates from all conjuncts.

[9] Unfortunately, a more detailed comparison – requested by a reviewer – of the current proposal to previous work of Sylvain Kahane, Nicolas Mazziotta and Kim Gerdes carried out within such combined approaches is impossible within the limits of this paper.

[10] Recall that the subordination relation is the reflexive transitive closure of the dependency relation.

Now, a representation of different nestings is possible in the enhanced graph if the second – and only the second – way of distributing dependencies among conjuncts is adopted, i.e., if dependents of a coordinate structure are represented as headed by each conjunct. This is illustrated with (39):

(39) Tom and [[Jerry and Spike] and Scooby-Doo]:



Since *Scooby-Doo* is a `conj` dependent of the coordinated *Jerry and Spike*, the enhanced representation adds the `conj` dependency from *Spike* to *Scooby-Doo*. However, while the coordination *Jerry and Spike and Scooby-Doo* is a dependent of *Tom*, there are no additional edges from *Tom* to any non-initial conjuncts within *Jerry and Spike and Scooby-Doo*.

Just as in the case of the proposal of the previous subsection, it may be demonstrated that, for any number of conjuncts, this representation distinguishes between different nestings. Here, we illustrate all possibilities for the case of 4 conjuncts – see Table 2. Unlike in the case of the initial proposal of the previous subsection, different nestings may be distinguished on the basis of a single level of representation – the enhanced graph.

It is important to realise that this solution only works if just one part of the general UD approach to distributing coordination in enhanced representation is adopted: what is distributed among conjuncts are dependencies *from* coordinate structures, not dependencies *to* coordinate structures. As is easy to verify, the other two possibilities fail already in the case of 3 conjuncts: adopting both parts, i.e., also distributing dependencies *to* coordinate structures, fails to distinguish (3) from (4), while adopting only the latter part fails to distinguish (2) from (3). This might be construed as an additional argument for modifying the general UD rules of distribution in coordinate structures: distributing dependencies targeting coordinate structure is sometimes (p.c. to members of UD community) considered potentially confusing for downstream applications relying on enhanced representations; e.g., in the case of (37), looking just at the dependencies headed by the verb *like* in (38) suggests that there are two separate direct objects. Allowing only for the distribution of dependencies *from* coordinate structures would remove this problem and make the current proposal just a straightforward application of general UD principles.

4 Conclusion

While nested coordination is currently a problem for Universal Dependencies, a number of more or less conservative modifications rectifying this situation are in sight, including the possibility to enrich dependency labels in a way that indicates the nestings involved (cf. §3.2). However, we propose two solutions which do not lead to such a – theoretically unbounded – proliferation of labels and which fully preserve the current UD representation of coordination at the level of basic trees. The theoretical advantage of the solution presented in §3.3 is that it encodes the intuition – common in various linguistic approaches – that, in a sense, each conjunct is a head of a coordinate structure, but its practical disadvantage is that the enhanced representation is very different from current UD representations of coordination. The solution in §3.4 is much more UD-conservative, as the enhanced representation is a superset of the basic tree and it is constructed in full compliance with already existing UD mechanisms. Adopting either of these solutions would remove this embarrassing flaw in the current version of Universal Dependencies.

Acknowledgements

For useful comments, which led to numerous improvements of this paper, we wish to thank Tim Osborne, anonymous SyntaxFest 2019 reviewers, and the audience of the 17th International Workshop on Treebanks and Linguistic Theory (University of Oslo, December 2018). Agnieszka Patejuk gratefully acknowledges the *Mobilność Plus* mobility grant awarded to her by the Polish Ministry of Science and Higher Education.

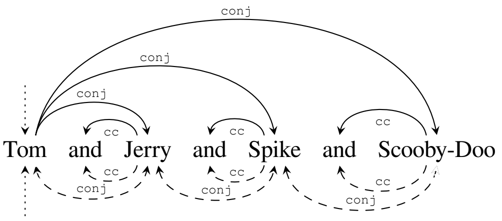
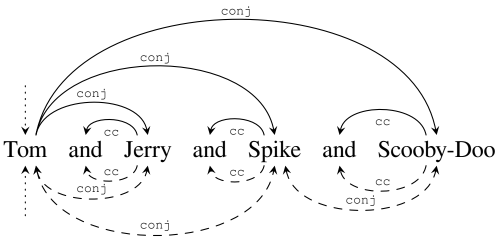
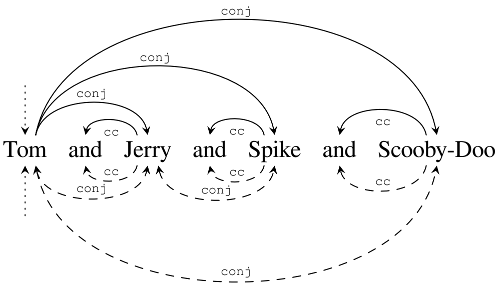
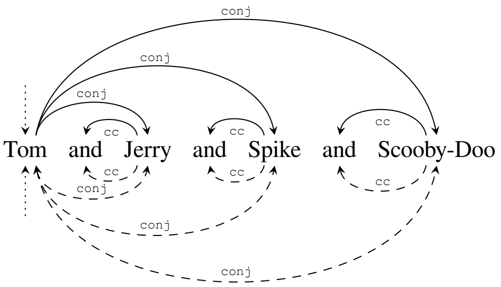
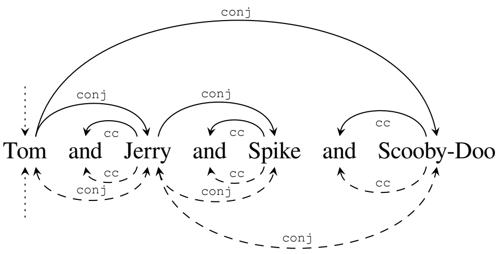
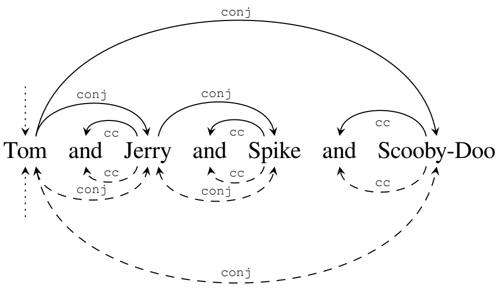
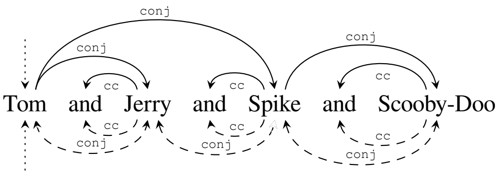
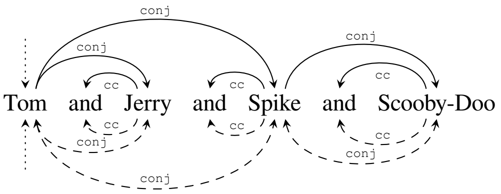
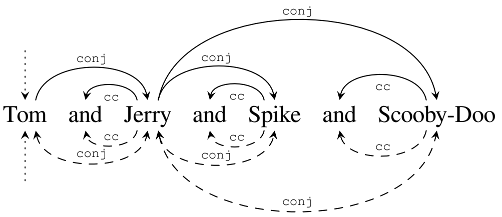
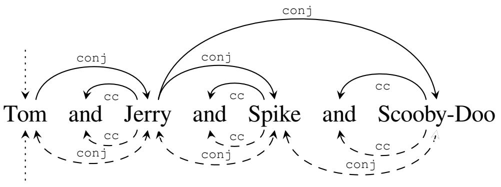
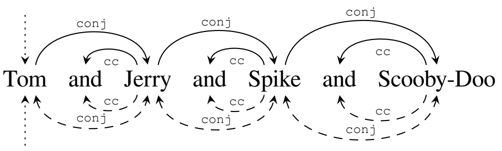
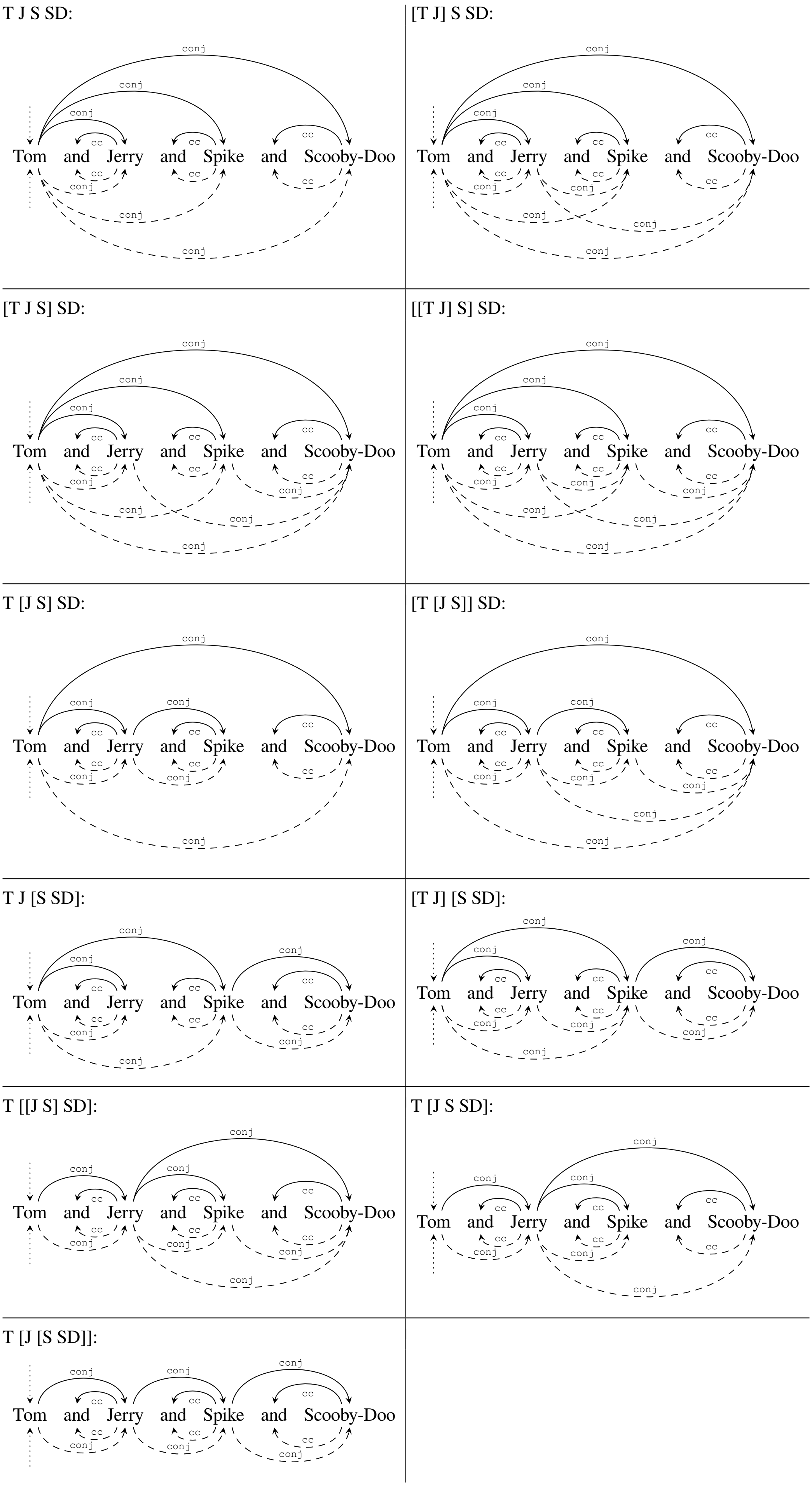
T J S SD: 	[T J] S SD: 
[T J S] SD: 	[[T J] S] SD: 
T [J S] SD: 	[T [J S]] SD: 
T J [S SD]: 	[T J] [S SD]: 
T [[J S] SD]: 	T [J S SD]: 
T [J [S SD]]: 	

Table 1: Possible nestings of *Tom and Jerry and Spike and Scooby-Doo* according to §3.3



T J S SD:

[T J] S SD:

[T J S] SD:

[[T J] S] SD:

T [J S] SD:

[T [J S]] SD:

T J [S SD]:

[T J] [S SD]:

T [[J S] SD]:

T [J S SD]:

T [J [S SD]]:

Table 2: Possible nestings of *Tom and Jerry and Spike and Scooby-Doo* according to §3.4

References

Anne Abeillé. 2003. A lexicon- and construction-based approach to coordinations. In Stefan Müller, editor, *Proceedings of the HPSG 2003 Conference*, pages 5–25, Stanford, CA: CSLI Publications.

Robert D. Borsley. 2005. Against ConjP. *Lingua* 115(4):461–482.

Gerald Gazdar, Ewan Klein, Goeffrey K. Pullum, and Ivan A. Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge / Cambridge, MA: Blackwell / Harvard University Press.

Kim Gerdes and Sylvain Kahane. 2015. Non-constituent coordination and other coordinative constructions as dependency graphs. In Eva Hajičová and Joakim Nivre, editors, *Proceedings of the Third International Conference on Dependency Linguistics (DepLing 2015)*, pages 101–110, Uppsala.

Richard Hudson. 1980. A second attack on constituency: A reply to Dahl. *Linguistics* 18:489–504.

Richard Hudson. 1984. *Word Grammar*. Oxford: Blackwell.

Richard Hudson. 1988. Coordination and grammatical relations. *Journal of Linguistics* 24(2):303–342.

Richard Hudson. 1989. Gapping and grammatical relations. *Journal of Linguistics* 25(1):57–94.

Richard Hudson. 1990. *English Word Grammar*. Oxford: Blackwell.

Richard Hudson. 2018. HPSG and Dependency Grammar, to appear in the HPSG handbook published by the Language Science Press.

Sylvain Kahane. 1997. Bubble trees and syntactic representations. In Tilman Becker and Hans-Ulrich Krieger, editors, *Proceedings of Mathematics of Language 5*, pages 70–76, DFKI, Saarbrücken.

Sylvain Kahane and Nicolas Mazziotta. 2015. Syntactic polygraphs: A formalism extending both constituency and dependency. In Marco Kuhlmann, Makoto Kanazawa, and Gregory M. Kobele, editors, *Proceedings of Mathematics of Language 14*, pages 152–164, Chicago, IL.

Hiroshi Kanayama, Na-Rae Han, Masayuki Asahara, Jena D. Hwang, Yusuke Miyao, Jinho D. Choi, and Yuji Matsumoto. 2018. Coordinate structures in Universal Dependencies for head-final languages. In Marie-Catherine de Marneffe, Teresa Lynn, and Sebastian Schuster, editors, *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 75–84, Association for Computational Linguistics.

Igor Mel'čuk. 1964. *Avtomatičeskij sintaksičeskij analiz. Tom I: Obščie principy. Vnutrisegmentnyj sintaksičeskij analiz*. Novosibirsk: Nauka.

Igor Mel'čuk. 1974. *Opyt teorii lingvističeskix modelej «Smysl ⇔ Tekst»*. Moscow: Nauka.

Igor Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. Albany, NY: The SUNY Press.

Igor Mel'čuk. 2009. Dependency in natural language. In Alain Polguère and Igor Mel'čuk, editors, *Dependency in Linguistic Description*, pages 1–110, Amsterdam: John Benjamins.

Igor Mel'čuk and Nikolaj Pertsov. 1987. *Surface Syntax of English. A Formal Model within the Meaning–Text Framework*. Amsterdam: John Benjamins.

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. 2016. Universal Dependencies v1: A multilingual treebank collection. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation, LREC 2016*, pages 1659–1666, ELRA, Portorož, Slovenia: European Language Resources Association (ELRA).

Timothy Osborne. 2006a. Gapping vs. non-gapping coordination. *Linguistische Berichte* 207:307–337.

Timothy Osborne. 2006b. Shared material and grammar: Toward a Dependency Grammar theory of non-gapping coordination for English and German. *Zeitschrift für Sprachwissenschaft* 25:39–93.

Timothy Osborne and Thomas Groß. 2017. Left node blocking. *Journal of Linguistics* 53(3):641–688.

Agnieszka Patejuk and Adam Przepiórkowski. 2018. *From Lexical Functional Grammar to Enhanced Universal Dependencies: Linguistically Informed Treebanks of Polish*. Warsaw: Institute of Computer Science, Polish Academy of Sciences.

Martin Pickering and Guy Barry. 1993. Dependency categorical grammar and coordination. *Linguistics* 31:855–902.

Carl Pollard and Ivan A. Sag. 1987. *Information-Based Syntax and Semantics, Volume 1: Fundamentals*. CSLI Lecture Notes, No. 13, Stanford, CA: CSLI Publications.

Carl Pollard and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago, IL: Chicago University Press / CSLI Publications.

Martin Popel, David Mareček, Jan Štěpánek, Daniel Zeman, and Zděněk Žabokrtský. 2013. Coordination structures in dependency treebanks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 517–527, Sofia, Bulgaria.

Adam Przepiórkowski and Agnieszka Patejuk. 2019. From Lexical Functional Grammar to enhanced Universal Dependencies: The UD-LFG treebank of Polish, to appear in *Language Resources and Evaluation* (published on-line on 4 February 2019).

Ivan A. Sag, Gerald Gazdar, Thomas Wasow, and Steven Weisler. 1985. Coordination and how to distinguish categories. *Natural Language and Linguistic Theory* 3:117–171.

Sebastian Schuster, Matthew Lamm, and Christopher D. Manning. 2017. Gapping constructions in Universal Dependencies v2. In Marie-Catherine de Marneffe, Joakim Nivre, and Sebastian Schuster, editors, *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 123–132, Association for Computational Linguistics, Gothenburg, Sweden.

Richard P. Stanley. 1997. Hipparchus, Plutarch, Schröder, and Hough. *The American Mathematical Monthly* 104(4):344–350.

Lucien Tesnière. 1959. *Éléments de Syntaxe Structurale*. Paris: Klincksieck.

Lucien Tesnière. 2015. *Elements of Structural Syntax*. Amsterdam: John Benjamins.

Universal Dependencies for Mbyá Guaraní

Guillaume Thomas
Department of Linguistics
University of Toronto
guillaume.thomas@utoronto.ca

Abstract

This paper presents the first treebank of Mbyá Guaraní, a Tupí-Guaraní language spoken in Argentina, Brazil and Paraguay. The Mbyá treebank is part of Universal Dependencies, a project that aims to create a set of guidelines for the consistent grammatical annotation of typologically different languages. We describe the composition of the treebank, and non-trivial choices that were made in the adaptation of Universal Dependencies guidelines to the annotation of Mbyá.

1 Introduction

Universal Dependencies (UD) is a cross-linguistic treebank annotation project, which aims to provide guidelines that are consistently applicable to typologically different languages (McDonald et al., 2013). Annotation guidelines are meant to be suitable for computer parsing, while enabling rigorous typological research and linguistic analysis of individual languages. They should also be easily understood by non-linguists. At the time of writing this paper, UD version 2.4 consists of 146 treebanks in 83 languages (Nivre et al., 2019).

This paper discusses the creation of a UD treebank for Mbyá Guaraní, a Tupí Guaraní language (Tupian) spoken in Argentina, Brazil and Paraguay. Work on indigenous American language in Universal Dependencies is still scarce. Previous research on the suitability of Universal Dependencies for the analysis of indigenous American languages include work on Arapaho, an Algonquian language (Wagner et al., 2016) and Shipibo-Konibo, a Panoan language (Vasquez et al., 2018). Outside of UD, Mikkelsen et al. (2014) discuss the development of a dependency treebank of Karuk, a isolate within the Hokan group. Our goals in this paper are to motivate the choices that were made in adapting UD guidelines for the annotation of Mbyá, and to reflect on difficulties that were encountered in this process. In doing so, we hope to contribute to the ongoing debate on the typological foundations of the UD project.

The treebank consists of two parts, each of which has been included in Universal Dependencies v2.4 (Thomas 2019a,b). The paper refers to the latest development version of the UD Mbyá Treebank at the time of writing. We assume familiarity with UD v2 guidelines, as described in UD Guidelines (n.d.).

2 General Information and Treebank Composition

Mbyá is a Guaraní language spoken by approximately 30,000 speakers in Argentina, Brazil and Paraguay (Ladeira, 2018). It belongs to the southern branch (group 1) of the Tupí Guaraní family, together with Nhandeva, Kaiowá and Paraguayan Guaraní, among other languages (Rodrigues, 1986). The main references on the grammar of Mbyá are Robert Dooley's grammatical sketch and lexicon of the language (Dooley, 2015), and Martins (2003)'s doctoral dissertation.

The UD Mbyá treebank consists of two corpora. The largest one is composed of narratives collected by Robert Dooley, written by two Mbyá Guaraní speakers, Nelson Florentino and Darci Pires de Lima, between 1976 and 1990 in Brazil. It contains 11,771 tokens (1,046 sentences). Interlinearized versions of these narratives are archived on the Archive of the Indigenous Languages of Latin America (Dooley, n.d.), and were used with Robert Dooley's authorization. The second corpus is composed of three speeches by Paulina Kerechu Núñez Romero, a Mbyá Guaraní speaker from Ytu community, Caazapá Department, Paraguay, which were recorded by the author. It consists of 1,318 tokens (98 sentences).

There is no standard orthography of Mbyá. Dooley's corpus uses the orthography presented in Dooley (2015), which is popular among Mbyá communities in the south of Brazil. The texts collected in Paraguay uses an adaptation of this orthography to Spanish based spelling conventions adopted in Mbyá communities in Argentina and Paraguay.

The texts were manually interlinearized in SIL Fieldworks Language Explorer (FLEx; Black and Simons (2008)). Robert Dooley's interlinearization of Florentino and Pires de Lima's narratives was imported into FLEx and revised to fit our annotation guidelines. Language specific parts of speech were added manually at this stage of annotation. Interlinearized narratives were exported in the XML FLEx-Text format from FLEx and converted to the CoNLL-U format using a Python script. Morphosyntactic features were automatically created in the conversion stage. Universal POS tags were automatically converted from language specific tags at this stage too, and were later corrected manually. Dependency annotation was semi-automatic. A first set of 500 sentences were annotated manually in Arborator (Gerdes, 2013), and was used to train a parser in UDPipe (Straka et al., 2016). This parser was used to annotate the rest of the corpus, which was manually corrected in Arborator. The annotation team consisted of the author and research assistants at the University of Toronto.[1]

3 Annotation Guidelines

Our annotation guidelines are based on version 2 of Universal Dependencies (UD Guidelines, n.d.). In this section, we describe the adaptation of UD guidelines to Mbyá, focusing on phenomena that are specific to the annotation of Mbyá or that raise interesting issues for the UD annotation scheme.

3.1 Lexical Categories

Universal Parts of Speech (POS) in UD include 6 POS for open class words: Adjectives (`ADJ`), Adverbs (`ADV`), Interjections (`INTJ`), Nouns (`NOUN`), Proper Nouns (`PROPN`) and Verbs (`VERB`). Here, we will focus on the distinction between `ADJ`, `ADV`, `NOUN` and `VERB`. While most scholars of Guaraní languages recognize the existence of a noun/verb distinction, there is less agreement on the existence of a distinction between adjectives and adverbs in these languages (see Dietrich (2017) for a recent discussion). Consequently, we have not included these categories in the language specific tagset for Mbyá. The following subsections describe the mapping from these language specific categories to the universal POS of UD.

Verbs The language specific tagset of Mbyá includes subcategories of verbs that reflect their valency and agreement class. In order to understand this categorization, it is necessary to give some background on agreement in the language. Subjects and objects are cross-referenced on verbs by prefixes that encode person and number. There are two sets of cross-reference prefixes, which I refer to as 'set A' and 'set B' prefixes, following Tonhauser (2017). These two sets distinguish two classes of intransitive verbs. Set A prefixes are used to index the subject of active (dynamic) verbs, while set B prefixes are used with inactive (stative) verbs, as illustrated by examples (1) and (2). Accordingly, we distinguish active (`vi:a`) from inactive (`vi:i`) intransitive verbs in our language specific tagset:[2]

(1) A-vaẽ.
A1.SG-arrive
`VERB`
`vi:a`
'*I arrived.*'

(2) Xe-kane'õ.
B1.SG-tired
`VERB`
`vi:i`
'*I am tired.*'

[1] Gregory Antono, Laurestine Bradford, Vidhyia Elango, Jean-François Juneau, Angelika Kiss, Barbara Peixoto, Darragh Winkelman.

[2] Glosses: A1.SG: first person singular 'active' inflection; A1.PL.INC: first person plural inclusive 'active' inflection; A1.PL.EXCL: first person plural exclusive 'active' inflection; B1.SG: first person singular 'inactive' inflection; BDY; information structure boundary marker; COMP: completive aspect; CONT: continuative aspect; DM discourse marker; DS different subject marker; HSY: hearsay evidential; MIR: mirative evidential; NEG negation; NMLZ nominalization; NPOSSD: non-possessed nominal form; PAST: past tense; R: linker morpheme; REFL: reflexive; REL: relativizer; SS same subject marker.

Words that were categorized as verbs in the language specific tagset were also tagged as `VERB` when they are used as predicates. In addition, inactive verbs (vi:i) are also attested as modifiers of nouns and of non-nominal heads, in which case they were tagged as (`ADJ`) or (`ADV`), respectively:

(3)

Avaxi	o-nhotỹ	r-yxy	porã.
Corn	A3-plant	R-line	good
`NOUN`	`VERB`	`NOUN`	`ADJ`
`n`	`vt`	`n`	`vi:i`

‘*He planted the corn in a good line*.’ (Dooley, 2015)

(4)

Oro-vy’a	porã.
A1.PL.EXCL-happy	good
`VERB`	`ADV`
`vi:a`	`vi:i`

‘*We were very happy*.’ (Dooley, 2015)

Alternatively, verbs tagged as `ADJ` or `ADV` could have been tagged as `VERB`, and analyzed as reduced relative or adverbial clauses. However, when used as modifiers, these verbs are typically uninflected (i.e. they do not bear cross-reference prefixes), unlike verbs in fully fledged clausal modifiers. Consequently, we believe that verb roots used as modifiers do not head a clause, which means that they should be tagged as `ADJ` or `ADV` in UD v2.4 guidelines.

Nouns Drawing a distinction between nouns and inactive verbs is not trivial in Mbyá, since the B set of cross-reference markers of inactive verbs is also used as possessive markers on nouns, and nouns are productively used as predicates without copula (for a discussion of this issue in Tupí-Guaraní languages, see Queixalós (2001)). Following Dooley (2015), we categorize as nouns those words that can be used as arguments without additional marking (such as nominalizing morphology). In order to preserve the distinction between nominal and verbal predications, these words were tagged as nouns both in their argument uses and in their predicative uses.

Adjectives and Adverbs Some roots can be used as modifiers but not as predicates. Because of the lack of evidence of a lexical distinction between adjectives and adverbs in the language, they were tagged as modifiers in the language specific tagset, and as `ADJ` or `ADV` in the universal tagset. This is illustrated by the use of *guaxu* (‘big’, ‘a lot’) in the following examples:

(5)

Ja-j-apo	karu	guaxu.
A1.PL.INCL-B3-do	meal	big
`VERB`	`NOUN`	`ADJ`
`vi:a`	`n`	`mod`

‘*We prepared a big meal*.’ (Dooley 2015)

(6)

A-karu	guaxu.
A1.SG-eat	a lot
`VERB`	`ADV`
`vi:a`	`mod`

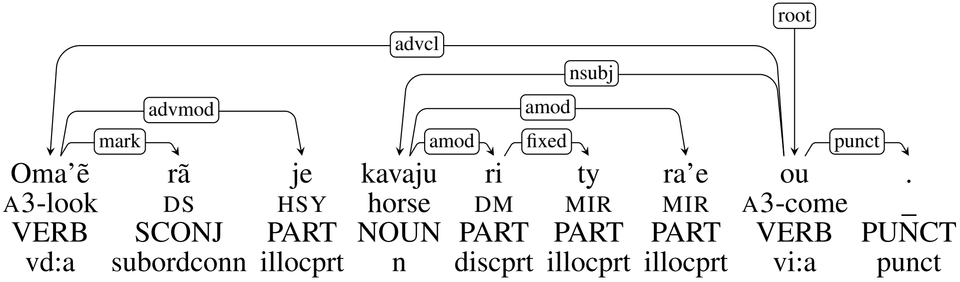
‘*I ate a lot*.’ (Dooley 2015)

3.2 Particles

Mbyá has a rich system of particles, which have been glossed as `PART` in the universal tagset. Language specific tags for particles encode their semantic function: aspect particles (`aspprt`), discourse particles `discprt`, focus particle (`focprt`), illocutionary particles (`illocprt`), intensifiers (`intprt`), modal particles (`modprt`), quantificational particles (`quantprt`), question particles (`qprt`) and temporal particles (`temprt`).

Many of these particles can be used as dependents of nouns as well as of verbs. This raises an issue for the UD v2 annotation scheme, since the only functional dependencies of nominal heads admitted by the guidelines are determiners (`det`), classifiers (`clf`) and case (`case`). The solution we have adopted is to default to `amod` for particles used as modifiers of nouns, as illustrated in example (7), where the mirative particles *ri ty* and *ra’e* modify the noun *kavaju* (‘horse’):

(7) ‘*When he looked, a horse had arrived.*’ (Dooley, n.d.)


root
advcl
nsubj
advmod
amod
mark
amod
fixed
punct
Oma’ẽ rã je kavaju ri ty ra’e ou .
A3-look DS HSY horse DM MIR MIR A3-come _
VERB SCONJ PART NOUN PART PART PART VERB PUNCT
vd:a subordconn illocprt n discprt illocprt illocprt vi:a punct

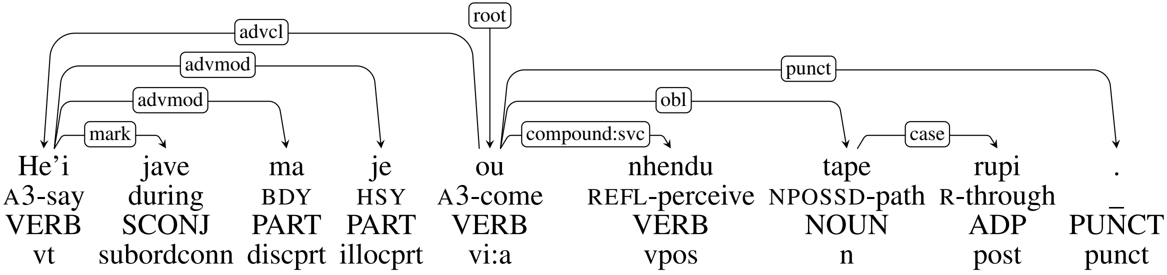
This solution, however, is unsatisfying, since particles are not adjectives, but belong to a closed class of functional items. A more satisfying solution would be to introduce a dependency relation for modifiers that is neutral with respect to the syntactic category of the dependent, as advocated by Croft et al. (2017).

Particles that modify non-nominal heads were annotated with the `advmod` relation, as illustrated by the hearsay evidential *je* in example (7). Alternatively, particles expressing tense, aspect, mood and evidentiality (TAME) might have been tagged as `AUX` when they modify a verb, in which case the relation `aux` would have been used. However, since TAME particles have no morphological verbal features, and are so flexible in their distribution, we are reluctant to tag them as auxiliaries. This being said, the semantic subcategorization of particles in the language specific tagset should make it trivial to map TAME particles to auxiliaries, when they are used as modifiers of verbs.

3.3 Complex Predicates

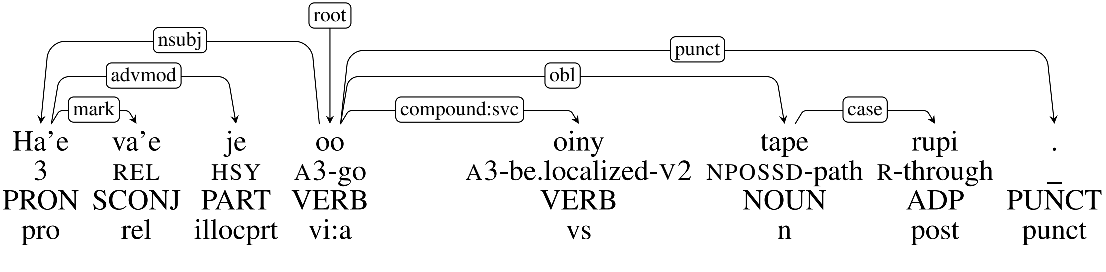
There are at least two types of complex predicates in Mbyá. The first one is formed by combining the main verb with a bare uninflected root, which we glossed `vpos` in the language specific POS tagset. Postposed roots are used in the expression of agent oriented modality (e.g. *pota*, ‘try to’), or sensory evidentiality, like *nhendu* (‘audibly’) in the following example:

(8) ‘*As she was saying this, she heard something coming on the road.*’ (Dooley, n.d.)


root
advcl
advmod
punct
advmod
obl
mark
compound:svc
case
He’i jave ma je ou nhendu tape rupi .
A3-say during BDY HSY A3-come REFL-perceive NPOSSD-path R-through _
VERB SCONJ PART PART VERB VERB NOUN ADP PUNCT
vt subordconn discprt illocprt vi:a vpos n post punct

A second type of complex predicates is formed by using one of a limited number of verbs in the so-called ‘gerund’ form common in Tupí-Guaraní languages (Rodrigues, 1953; Jensen, 1989), which we glossed `vs` in the language specific tagset. The dependent verb can be interpreted literally as expressing the position in which the event described by the main verb is realized, but it can also have an aspectual value. This construction was described by Dooley (1991):

(9) ‘*He was walking down the road.*’ (Dooley, n.d.)


root
nsubj
punct
advmod
obl
mark
compound:svc
case
Ha’e va’e je oo oiny tape rupi .
3 REL HSY A3-go A3-be.localized-V2 NPOSSD-path R-through _
PRON SCONJ PART VERB VERB NOUN ADP PUNCT
pro rel illocprt vi:a vs n post punct

Both types of complex predicates were annotated with the relation `compound:svc` in the treebank.

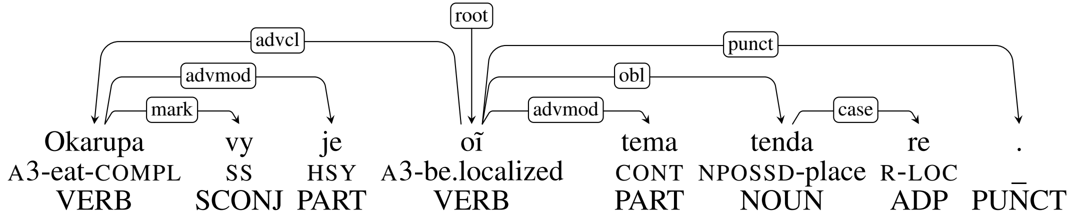
3.4 Strategies of Subordination

The UD v2 annotation scheme recognizes three major types of subordinate clauses: core clausal arguments, adverbial clauses and relative clauses. We describe each of these in turn in this section.

3.4.1 Adverbial Clauses

Adverbial clauses are introduced by a variety of subordinating conjunctions (`SCONJ`). Among these, it is useful to distinguish plain subordinating conjunctions from switch reference markers. The former, such as *jave* ('when'), only express temporal or causal relations between clauses. The switch reference markers *vy* and *ramo*/*rã* also function as subordinating conjunctions, but do not encode a specific temporal or causal relation. Instead, they indicate whether the subject of the subordinated clause is the same as that of the superordinate clause. In example (10), the same subject marker *vy* indicates that the subject of the eating is the same as the subject of the sitting:

(10) '*After he had eaten, he was sitting on a bench.*' (Dooley, n.d.)

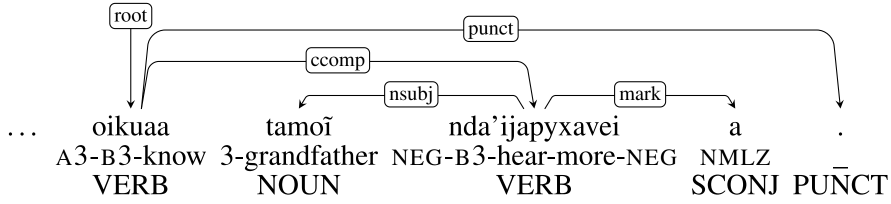


Our annotation guidelines treat switch reference markers as subordinating conjunctions, which are introduced by the relation `mark`.

3.4.2 Nominalized Complement Clauses

There are no subject clauses in the treebank. Complement clauses are attested and are formed by nominalizing the dependent verb with the morpheme *a*. While Dooley (2015) analyzes this morpheme as a suffix, we might argue that it is a clitic, since its host is not necessarily the verb, but can be one of its adverbial modifiers. This makes it unsatisfying to analyze the nominalizer as a suffix and to represent its function as a verbal feature, since in some cases this feature would have to appear on a word other than that to which the nominalizer is affixed. Consequently, we decided to represent this nominalizer as a token in the dependency annotation, where it is tagged as `SCONJ` and related to its head by a `mark` dependency, as illustrated in example (11). This decision is consistent with some writing conventions for Mbyá, such as those adopted by Cadogan (1959).

(11) '*He knew that his grandfather couldn't hear well anymore.*' (Dooley, n.d.)



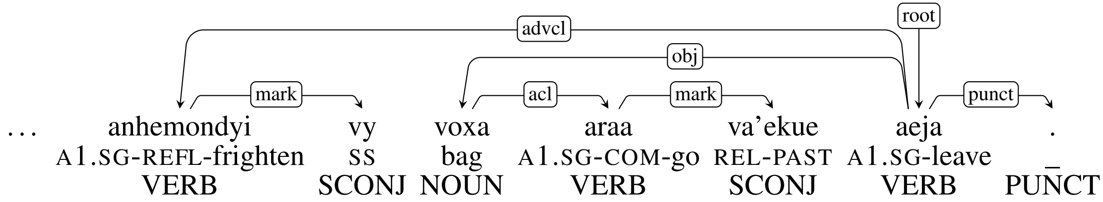
Nominalized clauses have several morphosyntactic features that are characteristic of noun phrases. They are compatible with temporal suffixes, which in Guaraní languages are nominal markers, and they may be used as complements of post-positions. Nevertheless, they preserve their full clausal structure. In particular, the verbs that head clausal nominalizations project their regular argument structure, bear cross-reference markers, may be modified by adverbs and may be part of complex predicates.

The mixed categorial status of clausal nominalizations in Mbyá raises the question of which dependency relation should be used to relate them to their head. In particular, should nominalized complements of verbs be introduced by the `ccomp` relation, or by the `obj` relation? Using the `ccomp` relation allows us to capture the fact that these complements are propositional. In this case, we take `ccomp` to indicate the semantic status of its dependent, a proposition rather than an individual. In addition, the use of `ccomp` indicates the fact that the complement has a full clausal structure. On the other hand, using `obj` is consistent with the nominal category of the complement, in particular its compatibility with adpositions, which UD represents as case marking of nominal dependents. In devising annotation guidelines for the Mbyá treebank, we have decided to use the `ccomp` relation with nominalized clauses that denote propositions.

3.4.3 Relative Clauses

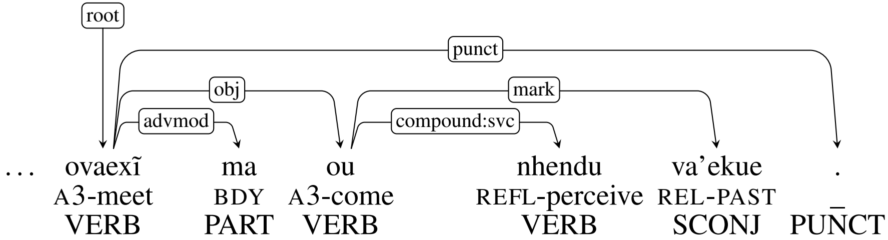
Relative clauses are formed with the enclitic *va'e*, as illustrated in example (12).

(12) '*I got scared, and I left a bag that I had brought.*' (Dooley, n.d.)


root
advcl
obj
mark
acl
mark
punct
... anhemondyi vy voxa araa va'ekue aeja .
A1.SG-REFL-frighten SS bag A1.SG-COM-go REL-PAST A1.SG-leave _
VERB SCONJ NOUN VERB SCONJ VERB PUNCT

Like the clitic *a*, *va'e* is a nominalizer, and relative clauses have a mixed syntactic category. This create an issue for the annotation of free relative clauses used as arguments of verbs, like *ou nhendu va'ekue* (lit. 'that he had heard coming') in the following example:

(13) '*He met the person that he had heard coming.*' (Dooley, n.d.)

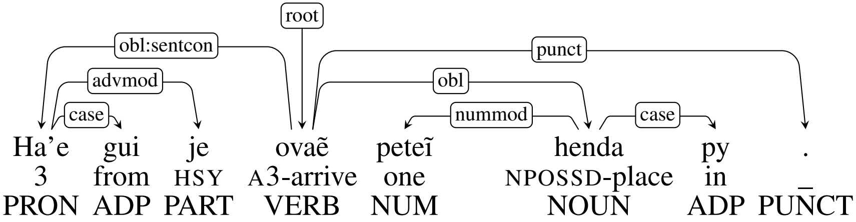

root
punct
obj
mark
advmod
compound:svc
... ovaexĩ ma ou nhendu va'ekue .
A3-meet BDY A3-come REFL-perceive REL-PAST _
VERB PART VERB VERB SCONJ PUNCT

In this example, the complement of *ovaexĩ* ('meet') is syntactically clausal, yet it denotes an individual. The first property motivates the use of a `ccomp` relation, while the second supports the use of an `obj` relation. We have opted for the latter, in order to capture the contrast between clausal nominalizations that denote individuals, introduced by `obj`, and those that denote propositions, introduced by `ccomp`.

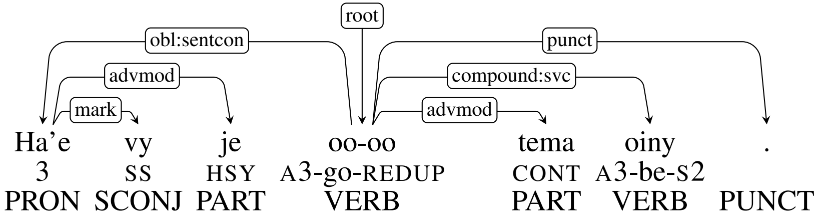
3.5 Discourse Connectives

Most sentences in narratives start with a sentence initial discourse connective. These connectives are composed of the pronoun *ha'e*, which is generally followed by an adposition or a subordinating conjunction, as illustrated in examples (14) and (15). Following Dooley (2015), we assume that occurrences of *ha'e* in discourse connectives denote propositions or situations that are described or made salient by a preceding discourse unit, much like the demonstrative *this* in the English connective *contrary to this*. Since their head is pronominal, we analyze sentence initial discourse connectives as oblique modifiers of the root:

(14) '*Finally, he arrived at a place.*' (Dooley, n.d.)

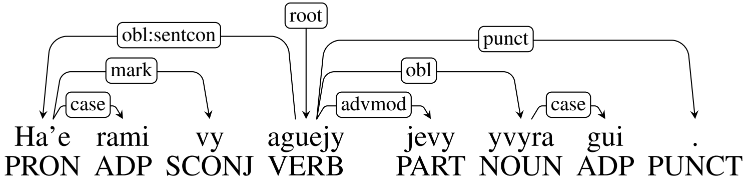

root
obl:sentcon
punct
advmod
obl
case
nummod
case
Ha'e gui je ovaẽ peteĩ henda py .
3 from HSY A3-arrive one NPOSSD-place in _
PRON ADP PART VERB NUM NOUN ADP PUNCT

(15) ‘*He walked and walked.*’ (Dooley, n.d.)


root
obl:sentcon
punct
advmod
compound:svc
mark
advmod
Ha’e vy je oo-oo tema oiny .
3 SS HSY A3-go-REDUP CONT A3-be-S2
PRON SCONJ PART VERB PART VERB PUNCT

Note that these propositional pronouns can be modified by an adposition and a subordinating expression simultaneously, as illustrated in (16):

(16) ‘*Because of this, I climbed down the tree again.*’ (Dooley, n.d.)


root
obl:sentcon
punct
mark
obl
case
advmod
case
Ha’e rami vy aguejy jevy yvyra gui .
PRON ADP SCONJ VERB PART NOUN ADP PUNCT

Sentence initial discourse connectives are another manifestation of the blurring of the distinction between clausal and nominal categories in Mbyá. Their heads are pronominal. As such, they are compatible with plural marking by the particle *kuery*, and they can be introduced by post-positions, which are characteristic features of nouns. On the other hand, their heads denote propositions, and are compatible with subordinating conjunctions like the same subject marker *vy* in example (15), which normally introduces adverbial clauses. In this example, *vy* indicates that the subject of the verb *oo* is the same as that of the previous sentence, which provides an antecedent to the pronoun *ha’e*.

We have decided to annotate sentence initial discourse connectives as obliques (`obl`) rather than adverbial clauses (`advcl`), thereby giving more weight to the form of their heads (pronominal) than to their interpretation (propositional).

4 Conclusion

We presented the Mbyá treebank, a syntactically annotated corpus of Mbyá in Universal Dependencies. We discussed the adaptation of UD guidelines to the annotation of Mbyá, highlighting questions raised by mixed categories (nominalizations) and the use of functional particles as adnominal modifiers.

Acknowledgements

Initial work on the treebank was made possible by a research award to the author by the Connaught Fund at the University of Toronto.

References

Andrew Black and Gary Simons. 2008. The SIL Fieldworks Language Explorer Approach to Morphological Parsing. In *Computational Linguistics for Less Studied Languages: Texas Linguistics Society, 10*, pages 37–55. CSLI Publications.

León Cadogan. 1992. *Ayvu Rapyta; Textos Míticos de los Mbyá Guaraní del Guairá.* Boletim n^{o} 227 – Antropología n^{o} 5. São Paulo: Universidade de São Paulo.

William Croft, Dawn Nordquist, Katherine Looney and Michael Regan. 2017. Linguistic typology meets Universal Dependencies. In Markus Dickinson, Jan Hajic, Sandra Kübler and Adam Przepiórkowski (eds), *Proceedings of the 15th International Workshop on Treebanks and Linguistic Theories* (TLT15), pages 63–75. CEUR Workshop Proceedings

Wolf Dietrich. 2017. Word Classes and Word Class Switching in Guaraní Syntax. In Bruno Estigarribia and Justin Pinta (eds), *Guaraní Linguistics in the 21st century*, pages 158–193. Leiden: Brill.

Robert A. Dooley. 1991. A double-verb construction in Mbyá Guaraní. In *Work Papers of the Summer Institute of Linguistics, University of North Dakota Session* 35:31–66.

Robert A. Dooley. 2015. *Léxico guarani, dialeto mbyá*. Summer Institute of Linguistics.

Robert A. Dooley. n.d. *Mbyá Guaraní collection of Robert Dooley*. The Archive of the Indigenous Languages of Latin America: www.ailla.utexas.org. Media: text. Access: 100

Kim Gerdes. 2013. Collaborative dependency annotation. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 88–97.

Cheryl Jensen. 1989. *O desenvolvimento histórico da língua Wayampi.* Campinas: Editora da UNICAMP.

Maria Inês Ladeira. 2018. Guarani Mbya. Em Fany Pantaleoni Ricardo (ed.), *Povos Indígenas no Brasil*, Instituto Socioambiental. `https://pib.socioambiental.org/pt/Povo:Guarani_Mbya`

Marci Fileti Martins. 2003. *Descrição e análise de aspectos de gramática do guarani mbyá [Description and analysis of some grammatical aspects of Guaraní Mbyá].* Ph.D. thesis, State University of Campinas.

Line Mikkelsen, Andrew Garrett, Erik Maier and Clare Sandy. 2014. Developing a syntactically parsed corpus of Karuk. Talk presented at the Annual Meeting of the Society for the Study of the Indigenous Languages of the Americas, Minneapolis, MN. January 3rd.

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

Joakim Nivre, Mitchell Abrams, Agić Željko et al. 2019. *Universal dependencies 2.4*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Francesc Queixalós (ed.) 2001. Des noms et des verbes en tupi-guarani: état de la question. LINCOM Studies in Native American Linguistics, vol. 37. Munich: LINCOM Europa.

Aryon Dall'Igna Rodrigues. 1953 Morfologia do verbo tupí. *Letras*, 1:121–152.

Aryon Dall'Igna Rodrigues. 1986. *Línguas brasileiras*. São Paulo: Loyola.

Milan Straka, Jan Hajič and Jana Straková. 2016. UD-Pipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation* (LREC'16).

Guillaume Thomas. 2019a. UD Mbya_Guarani_Dooley, Mbyá Guaraní treebank based on narratives collected by Robert Dooley. In Nivre et al. 2019.

Guillaume Thomas. 2019b. UD Mbya_Guarani_Thomas, Mbyá Guaraní treebank based on narratives collected by Guillaume Thomas. In Nivre et al. 2019.

Judith Tonhauser. 2017. The Distribution of Implicit Arguments in Paraguayan Guaraní. In Bruno Estigarribia and Justin Pinta (eds), *Guaraní Linguistics in the 21st century*, pages 231–258. Leiden: Brill.

Alonso Vasquez, Renzo Ego Aguirre, Candy Angulo, John Miller, Claudia Villanueva, Željko Agić, Roberto Zariquiey and Arturo Oncevay. 2018. Toward Universal Dependencies for Shipibo-Konibo. Proceedings of the Second Workshop on Universal Dependencies (UDW 2018), pages 151–161. `https://www.aclweb.org/anthology/W18-6018`

Irina Wagner, Andrew Cowell and Jena Hwang. 2016. Applying Universal Dependency to the Arapaho Language. Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016), pages 171–179. DOI: 10.18653/v1/W16-1719.

Universal Dependencies n.d. Universal Dependencies Guidelines. `https://universaldependencies.org/guidelines.html`

Survey of Uralic Universal Dependencies development

Niko Partanen
University of Helsinki
niko.partanen@helsinki.fi

Jack Rueter
University of Helsinki
jack.rueter@helsinki.fi

Abstract

This paper attempts to evaluate some of the systematic differences in Uralic Universal Dependencies treebanks from a perspective that would help to introduce reasonable improvements in treebank annotation consistency within this language family. The study finds that the coverage of Uralic languages in the project is already relatively high, and the majority of typically Uralic features are already present and can be discussed on the basis of existing treebanks. Some of the idiosyncrasies found in individual treebanks stem from language-internal grammar traditions, and could be a target for harmonization in later phases.

1 Introduction

The Uralic languages constitute one of the major language families in the world. There are approximately 38 languages represented by seven major branches in the family tree. Only Finnish, Estonian and Hungarian are majority languages in their states, and other Uralic languages are minority languages, often endangered, in their respective regions, including Northern Scandinavia and Russia.

Uralic languages are agglutinative, morphologically rich and typically have large case systems. The majority of Uralic languages share many features, including, the expression of negation with a verb of negation, preference of postpositions and complex object marking. They usually also have a complex system of non-finite forms. Constituent order is relatively flexible. However, since the different branches of the Uralic family are rather far removed from one another, there are also numerous independent reflexes of historically shared features on the individual language level.

Recent versions of Universal Dependencies treebanks (Nivre et al., 2019) include 11 treebanks in seven Uralic languages. All in all, they represent five of the aforementioned major branches in the family. In release 2.4 the Uralic languages include:

- Erzya, 1 treebank (Rueter and Tyers, 2018)
- Estonian, 2 treebanks (Muischnek et al., 2014), (Muischnek et al., 2016)
- Finnish, 3 treebanks (Haverinen et al., 2014), (Pyysalo et al., 2015)
- Hungarian, 1 treebank (Vincze et al., 2017)
- Karelian, 1 treebank (Pirinen, 2019)
- Komi-Zyrian, 2 treebanks (Partanen et al., 2018)
- North Sámi, 1 treebank (Sheyanova and Tyers, 2017)

In addition to the languages listed above, there are at least plans for a Northern Mansi treebank outlined in the literature (Horváth et al., 2017, 63), and work for Livvi Karelian treebank is undergoing. At the moment, the branches of the Uralic language family that do not have a single treebank are Samoyed and Mari, while the Ugric branch is only represented by Hungarian, missing the Ob-Ugric languages. From this point of view, the Northern Mansi treebank mentioned would be a most welcome addition. This would also bring improvement to the geographical limitations in the current selection of languages, where all now available treebanks represent Uralic languages spoken in Europe, with no language spoken primarily in Siberia.

Finite-State Transducers, especially in the Giellatekno infrastructure (Moshagen et al., 2014), have traditionally played an important role in open-source language technology for Uralic languages. Recent work in harmonizing NLP solutions (Hämäläinen, 2019) and lexical resources (Hämäläinen and Rueter, 2018) in Uralic languages may also prove beneficial for these efforts. Many of the Uralic treebanks have been created by automatic conversion of annotation schemes and tagsets from these analysers. Since several language documentation projects have also started to integrate these tools into their workflows (Gerstenberger et al., 2016), it would be expected that the pipelines used thus far could be reused when creating new treebanks for languages that have comparable resources.

Variation in annotation schemes has already been identified as one problem between analysers of different Uralic languages, and an earlier study by Tyers and Pirinen (2016) which examined this in detail. To continue research along these lines, we reproduce their Table 2 with minor corrections, and compare how the same examples align with the solution currently found in UD treebanks. This is one method to measure whether the common guidelines proposed in Tyers and Pirinen (2016) have taken root.

2 Selected Uralic features

In this section we go through selected features in the Universal Dependencies annotation scheme that can be commented specifically from a Uralic point of view.

2.1 Feature Evident

Komi-Zyrian has morphologically marked evidential forms of the verbs, and the Komi treebanks also use this feature with the value 'Evident=Nfh' to mark second past tense forms as non-firsthand information, although evidentiality is not there as an obligatory category as in some other languages, used primarily in unwitnessed narrative or to express non-voluntary action (Leinonen, 2000). Various evidentiality related phenomena occur in the morphology of other Permic languages, Mari and Ob-Ugric, and the Samoyedic languages, which, as mentioned, are still largely missing from the UD project. It can be stated that evidentiality is one feature for which Uralic family still has much to add in the project, although in the currently included languages it does not play a central role. The Erzya treebank, it will be noted, uses the 'Evident=Nfh' feature for some particles connected as advmod dependencies. Erzya treebank usage may open the discussion of introducing this feature in relation to other free word forms in the UD project. The Estonian treebank UD_Estonian-EDT, for example, does not have the feature Evident, although the concept is implicitly present in 'Mood=Qot'.

2.2 The Feature Gender

The Uralic languages do not have grammatical gender, as it were, permeating the pronoun, noun and verb systems. They do, however, have peripheral derivational elements, which are not regularly addressed, e.g. Finnish *-tar/-tär*: *ruhtinas* 'duke' vs *ruhtinatar* 'duchess', *näyttelijä* 'actor' vs *näyttelijätär* 'actress', etc. Erzya also a peripheral derivational element: *-низэ/-нызэ -ńize/-nïze*, which, traditionally, was added to the husband's name for indicating 'wife of': *Иван Ivan* vs. *Иваннызэ Ivannïze*, *Гава Gava* vs. *Гаванизэ Gavańize*. A similar construction predominantly occurs in the Hungarian language *né* 'wife of': *István* vs. *Istvánné*, but it is not found in the Hungarian treebank. Under normal circumstances, there should be no reason to mark gender feature in Uralic treebanks. One exception could arise in situations where the treebank contains code-switch speech to Russian, under which circumstances gender marking may be present (Janurik, 2015).

2.3 The Features Animacy and Definite

Animacy is not a grammatical category in Uralic languages, but it does influence the object marking in some languages within the family, e.g. in Komi, animacy has been connected to the marking of definiteness and focus as briefly described by Fediunova (2000, 69). At present, neither animacy nor definiteness have been marked in Komi-Zyrian treebanks, but definiteness can, in principle, be deduced from possessive suffixes used to this end. In the Erzya treebank, definiteness is marked as an incremental feature of the NP head morphology – similar to Scandinavian languages – yet it is distinct from the use of possessive marking. Hungarian also uses this feature, but it is collocated with the definite articles found in the language.

2.4 The Feature Aspect

At the moment, some Erzya verbs are marked with 'Aspect=Inch', and some Hungarian verbs are marked with 'Aspect=Iter'. Neither language has gone beyond expressing features for specific derivation morphology. In the Northern Saami treebank, however, 'Aspect' is used as a means for encoding participles, i.e. the perfect participle is coded with 'Aspect=Perf', whereas the present participle is coded with 'Tense=Present'. Neither Finnish, Estonian nor Komi use the Aspect feature at this point.

2.5 The Feature Number

Among the current Uralic treebanks, Northern Saami is the only one that has a dual number. The numbers used throughout are singular and plural. On the subject of number, however, there are several types to keep track of: simple *Number* is used with nominals to indicate the number of entities, *Number[Psor]* the possessor number, of course, tells us of the possessor flagged by possessive suffixes. When we arrive on the verb scene, Erzya introduces counting subject entities flagged on the finite verb with *Number[Subj]*, and object entities as well *Number[Obj]*. Hungarian introduces counting of possessa/possessee with *Number[Psee]* (see also (Vincze et al., 2017)). This is useful in Hungarian and could be feasible in any up-coming of treebank for Moksha, as well, e.g. Hungarian *kutya* 'dog' vs. *kutyáé* 'the one belonging to the dog', Moksha *пине* pińe 'dog' vs. *пиненне* pińeńńe 'one belonging to a dog'.

The 'Number' strategy sets a precedence for analogical regular inflectional features in Erzya and Komi-Zyrian. Where Erzya uses some of the oblique cases at both the NP level and the VP level, Komi has an operating dichotomy that distinguishes the two levels. If, for instance, the inessive Erzya *вирьсэ* viŕ-se 'in the forest' (derived from *вирь* viŕ 'forest') is taken as a premodifier in a noun phrase, Erzya morphology allows for constructions where NP head morphology is directly concatenated onto the premodifier, which might result in a form *вирьсэтнесэ* viŕ-se-t́-ńe-se 'in the ones that are in the forest' (a matter of ellipsis or 'secondary declension' as it is also refered to in the literature. Komi-Zyrian can derive a premodifier with the same semantics of its Erzya counterpart in *вöрса vərsa* 'in the forest' from *вöр* vər 'forest', which can in turn, as a NP head, take on either copula plural morphology (*вöрсаöсь* vər-sa-əɕ '[are] ones in the forest') or noun plural morphology (*вöрсаяс* vər-sa-jas 'the ones in the forest'). Although this regular morphology for Komi premodifiers is not addressed as case morphology in the largest of Komi grammars, (Fediunova, 2000), it merits contemplation in any extensive and parallel treatment of the language family. The fact that a second plural form can be present introduces further problems.

The Komi NP premodifier derivation strategies allow for plural stems. Hence, forms, such as *вöръяссаяс* vər-jas-sa-jas 'the ones that are in the forests' and *вöръяссаöсь* vər-jas-sa-əɕ '[are] the ones that are in the forests' may require regular counting and therefor a new 'Number': one to express the number of the NP head and the other indicate the number of the NP premodifier. In Komi, the locative *-са* -sa has further siblings in a temporal *-ся* -ɕa, a privative *-тöм* -təm and a proprietive *-а* -a, which should not be confused with inessive *-ын* -ɨn, caritive *-тöг* -təg, comitative *-кöд* -kəd or instrumental *-öн* -ən cases.

The situation seems to be similar to what is found in the Turkic languages, and the solution proposed in Çöltekin (2016) to split these words in Turkish into multiple tokens, unless they are lexicalized, would also be possible with the Uralic languages. The fact that Uralic languages do not have separate morphology analogic to the *ki* element in Turkic, however, would seem to speak against following such a lead.

The system of number marking outlined above seems to be a good starting point for all Uralic languages. It also sets the scene for a new discussion, which might draw from other morphologically rich language families and their practices of grammar description.

2.6 The Feature Case

At the moment all Uralic treebanks use traditional terms from their own grammars. Some of the terms, for example 'superessive', 'sublative' and 'additive' are only used in individual languages. On the one hand, this is understandable, but it begs the question as to how useful these terms actually are. Should these names indicate functions for a given language or should they be generalized. At least, in regard to spatial cases, the cross-lingual comparability is now rather weak. Some of the cases such as 'terminative', however, are already present in numerous treebanks and refer to a very similar concept. Similarly, cases like 'approximative', although currently present only in the Komi-Zyrian treebank, will eventually be wider present once languages with this case, such as Olonets-Karelian (Livvi), Ludic or Veps, are included.

Are the names of cases with multiple divergent functions relevant? The case names in Uralic languages are often very language-specific and are not transparent. Finnish derivation practice includes a mere mnemonic letter string to indicate a derivation, e.g. 'Derivation=Sti', which is basically a morphological representation for deadjectival adverb derivation. Without language-specific documentation neither case names nor derivation letter strings are meaningful. How to construct documentation in the manner that allows cross-linguistic comparison is a forthcoming challenge for treebank developers. Discussions should also include analogical solutions used in other language families, such as the English 'in' preposition equivalent of Finnish 'inessive' (Germanic prepositions are not generally given names for mapping them to equivalent Finnish case functions).

3 Selected syntactic questions

In this section we discuss some of the observations that can be made about the use of specific dependency relations, and questions that arise from morpho-syntactic particularities of Erzya and Komi.

3.1 The Dependency iobj

At the moment only the Hungarian treebank uses the dependency relation 'iobj'. In other Uralic treebanks, however, the relation 'obl' is used, which is illustrative of the examples shown in UD documentation for NPs with prepositional construction closely related to 'iobj'.[1] These present languages do not have dative alternation, but any future work with Mansi or Khanty may introduce this variation. It would be reasonable to assume that any updating of the version *2.2* Hungarian treebank would involve the introduction of the 'obl' relation where the 'iobj' is now used, perhaps with a special relation subtype of 'obl'.

In order to compare the question better and to illustrate the situation, we translated the English and French example sentence from UD documentation into different Uralic languages.

- English: give the children the toys & give the toys to the children
- French: donner les jouets aux enfants
- Estonian: andma mänguasju lastele
- Hungarian: a játékokat a gyerekeknek adja
- Finnish: antaa lapsille leluja
- Erzya: максомс налкшкетнень эйкакштнэнень
- Komi: сетны ворсанторъяс челядьлы

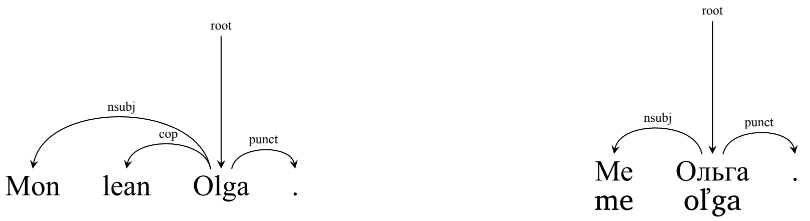
All example sentences above with the exception of the first English sentence are or (in the case of Hungarian) can be coded with the 'obl' relation due to the explicit morphological encoding of the NP head, which distinguishes, among others, the Hungarian dative case.

3.2 Copulas

The non-past identity clause involves copula morphology in all but the Komi language. While North Sámi, Estonian, Hungarian, Finnish and Karelian all have free copulas, Erzya provides examples of dependent morphology, which have elicited segmentation (locus + copula), on the one hand, and a discussion of word order issues, on the other.

The North Sámi in figure 1a is representative of word ordering typifying Finnish, Estonian and Karelian, alike – CS COPULA CC. Komi, however, does not use a COPULA, instead, it applies juxtaposition to achieve the same (see 1b).

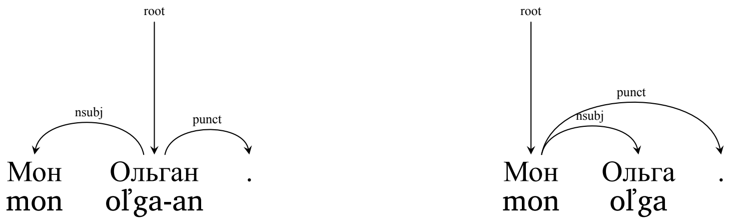
[1] https://universaldependencies.org/u/dep/all.html#al-u-dep/obl



(a) I am Olga. (North Sámi)

(b) I am Olga. (Komi)

Figure 1: Example with and without copula



(a) I'm OLGA. OR My name is OLGA. (Erzya)

(b) I'm Olga. OR My name's Olga. (Erzya)

Figure 2: Distinguishing Erzya Subject

In figures (2a) and (2b), distinctive functions in Erzya morpho-syntax are presented where first and second person personal pronouns can visibly serve as both copula subjects and copula complements. The dependent copula morphology has been split off of the root in the analysis (analogic to what has been applied in the UD_Turkish-IMST treebank), but when the analysis is non-past third person singular, i.e. ZERO, no extra token is introduced (cf. Tyers and Pirinen (2016) and Vincze et al. (2017)). The logic of the split solution in Erzya can be questioned. This question is underlined by the fact that Komi-Zyrian has copula complement plural marking *-öсь* -əɕ , which is used for marking attribution, location and even possession non-verbal predications non-past in much the same way as the more elaborate Erzya morphology – Komi does not split this affix off from the stem. If the copula morphology is segmented, the *-an* affix in figure (2a) is better illustrated by figure (3).

The distinction between (3) and (2b) lies in which argument the attributed agreement marking is. In (2b) it is the name that commands subject correlation, and prosodic stress falls on the personal pronoun root. In (3), however, the constant is actually the personal pronoun, and prosodic stress falls on the proper name root. These are matters, of course, for future work at discourse levels.

Unlike Turkic languages, the Erzya language has no unquestionable, distinct morphological element representing the copula in dependent marking other than what actually expresses tense, person and number. Although comparative linguistics does postulate the merging of a form of copula into the copula complement. For this reason, we are presented with a choice of following the Turkic lead, i.e. separating copula morphology from nouns (in numerous cases), adjectives and numerals for the soul purpose of reusing a ready solution. The non-past, third-person singular form of the copula complement, however, takes zero marking, which would point to non-symmetric representation of the copula construction. This, in turn, indicates a further need for a more elegant UD resolution of the issue.

4 Deverbal words and features

In Table 1 we utilize, correct and expand upon working knowledge in Uralic deverbal word constructions as represented in or analogous to UD work (cf. Tyers and Pirinen (2016)). Komi-Zyrian uses two finite clauses in sentence (i), where other languages all have a non-finite solution to the problem.

In sentence (ii), it appears that Hungarian and Estonian treat their deverbal nouns as nouns, while the other languages all encode their analogic forms with a spectrum of non-finite interpretations Conv, Ger, Inf and Sup. North Sámi, Erzya and Finnish mark person on the converb, which might not be mandatory.

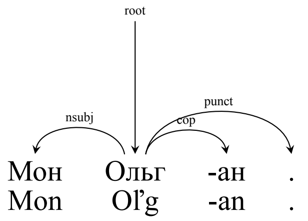


Figure 3: I’m OLGA. (Erzya)

<table>
<tr><th>Language</th><th>Sentence</th><th>UD</th></tr>
<tr><td>(i)</td><td>‘I saw the man running’</td><td></td></tr>
<tr><td>North Sámi</td><td>Oidnen dievddu <b>viehkame</b></td><td>Case=Ess|VerbForm=Ger</td></tr>
<tr><td>Erzya</td><td>Неия цёранть <b>чийнемадо</b>.</td><td>Case=Abl|Definite=Ind|Number=Plur,Sing|Valency=1|VerbForm=Vnoun</td></tr>
<tr><td>Finnish</td><td>Näin miehen <b>juoksemassa</b>.</td><td>Case=Ine|InfForm=3|VerbForm=Inf|Voice=Act</td></tr>
<tr><td>Estonian</td><td>Nägin meest <b>jooksmas</b>.</td><td>Case=Ine|VerbForm=Sup|Voice=Act</td></tr>
<tr><td>Hungarian</td><td>Láttam az embert <b>futni</b></td><td>VerbForm=Inf|Voice=Act</td></tr>
<tr><td>Komi-Zyrian</td><td>Аддзи, мортыс <b>котӧртӧ</b>.</td><td>Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin</td></tr>
<tr><td>(ii)</td><td>‘While running I saw the man’</td><td></td></tr>
<tr><td>North Sámi</td><td>Oidnen dievddu <b>viegadettiinan</b>.</td><td>Number[psor]=Sing|Person[psor]=1|VerbForm=Ger</td></tr>
<tr><td>Erzya</td><td><b>Чийнемстэнь</b> неия цёранть.</td><td>Case=Ela|Number[psor]=Sing|Person[psor]=1|VerbForm=Conv</td></tr>
<tr><td>Finnish</td><td>Näin miehen <b>juostessani</b>.</td><td>Case=Ine|InfForm=2|Number[psor]=Sing|Person[psor]=1|VerbForm=Inf|Voice=Act</td></tr>
<tr><td>Estonian</td><td><b>Jooksmise</b> ajal nägin ma meest.</td><td>Case=Gen|Number=Sing</td></tr>
<tr><td>Hungarian</td><td><b>Futás</b> közben láttam az embert.</td><td>‘NOUN’ _</td></tr>
<tr><td>Komi-Zyrian</td><td><b>Котралігӧн</b> аддзи мортӧс.</td><td>Case=Ins|Derivation=Ig|Number=Sing|VerbForm=Conv</td></tr>
<tr><td>(iii)</td><td>‘I see the running man.’</td><td></td></tr>
<tr><td>North Sámi</td><td>Oainnán <b>viehkki</b> dievddu.</td><td>Tense=Pres|VerbForm=Part</td></tr>
<tr><td>Erzya</td><td>Неян <b>чийниця</b> цёранть.</td><td>Case=Nom|Definite=Ind|Number=Sing|Tense=Pres|VerbForm=Part</td></tr>
<tr><td>Finnish</td><td>Näen <b>juoksevan</b> miehen.</td><td>Case=Gen|Number=Sing|PartForm=Pres|VerbForm=Part|Voice=Act</td></tr>
<tr><td>Estonian</td><td>Näen <b>jooksvat</b> meest.</td><td>Case=Par|Degree=Pos|Number=Sing|Tense=Pres|VerbForm=Part|Voice=Act</td></tr>
<tr><td>Hungarian</td><td>Látom a <b>futó</b> embert.</td><td>‘ADJ’ _</td></tr>
<tr><td>Komi-Zyrian</td><td>Аддза <b>котралысь</b> мортӧс.</td><td>PartForm=Pres|VerbForm=Part|Voice=Act</td></tr>
<tr><td>(iv)</td><td>‘Running is fun.’</td><td></td></tr>
<tr><td>North Sámi</td><td><b>Viehkan</b> lea suohtas.</td><td>Case=Nom|Number=Sing</td></tr>
<tr><td>Erzya</td><td><b>Чийнемась</b> вадря тев.</td><td>Case=Nom|Definite=Def|Number=Sing|VerbForm=Vnoun</td></tr>
<tr><td>Finnish</td><td><b>Juokseminen</b> on kivaa.</td><td>Case=Nom|Number=Sing</td></tr>
<tr><td>Estonian</td><td><b>Jooksmine</b> on lahe.</td><td>Case=Nom|Number=Sing</td></tr>
<tr><td>Hungarian</td><td>A <b>futás</b> jó dolog.</td><td>‘NOUN’ _</td></tr>
<tr><td>Komi-Zyrian</td><td><b>Котравны</b> лӧсьыд.</td><td>Case=Nom|Number=Sing|Tense=Past|VerbForm=Part</td></tr>
<tr><td>(v)</td><td>‘I like running.’</td><td></td></tr>
<tr><td>North Sámi</td><td>Liikon <b>viehkat</b>.</td><td>VerbForm=Inf</td></tr>
<tr><td>Erzya</td><td>Вечкса <b>чийнемам</b>.</td><td>Case=Gen|Number=Sing|Number[psor]=Sing|Person[psor]=1|Valency=2|VerbForm=Vnoun</td></tr>
<tr><td>Finnish</td><td>Pidän <b>juoksemisesta</b>.</td><td>Case=Ela|Number=Sing</td></tr>
<tr><td>Estonian</td><td>Mulle meeldib <b>joosta</b>.</td><td>VerbForm=Inf</td></tr>
<tr><td>Hungarian</td><td>Szeretek <b>futni</b>.</td><td>VerbForm=Inf|Voice=Act</td></tr>
<tr><td>Komi-Zyrian</td><td>Меным кажитчӧ <b>котралӧм</b>.</td><td>Case=Nom|Number=Sing|Tense=Past|VerbForm=Part</td></tr>
</table>

Table 1: Table reproduced and adapted from Tyers and Pirinen (2016, p. 98)

The present participle in sentence (iii) is treated in all treebanks, except for Hungarian, as a deverbal form. Komi-Zyrian and Finnish deviate from these by introducing a ‘PartForm=Pres’ value, deviating from the ‘Tense=Pres’ strategy of the other treebanks.

In sentence (iv) the subject in nearly all languages is regularly a deverbal noun, although derivation or inflection is not indicated in the Hungarian treebank. While North Sámi, Finnish and Estonian use an attributive copula construction, Erzya and Hungarian apply an equation construction with a noun head, and Komi-Zyrian uses a simple infinitive to mark the primary argument with an attributive copula construction.

Sentence (v) provides multiple solutions for the second argument of the matrix verb. While the secondary argument in North Sámi, Hungarian and Estonian is an infinitive, the other languages use a deverbal noun. It should be noted that the deverbal noun is an object in Erzya, a subject in Komi-Zyrian and an oblique in Finnish. The subject–object dichotomy may be observed in the use of infinitives, as well.

5 Summary

At the moment, Uralic treebanks are mainly representative of the largest languages in each branch, although with the recent addition of a Karelian treebank the coverage is already spreading in the direction of smaller Finnic languages with a high representation of Balto-Finnic languages. In the same vein, smaller Sámi languages would be very welcome to UD, and similarly Udmurt and Moksha would increase the diversity of these branches. As mentioned previously, Mari and Samoyed treebanks are still missing entirely, although some of these languages already have openly licensed annotated corpora that could easily be extended into treebanks. In principle the large coverage of Uralic languages at this point makes it realistic to expect that new treebanks would not introduce entirely different phenomena from what is already represented by the current Uralic languages. In the case of smaller and less studied Samoyedic languages, however, there may be questions that need specific attention, e.g. the expression of evidentiality and mood. Similarly Ob-Ugric languages may introduce dative alternation in Uralic languages.

The most common inconsistencies between languages in the Uralic treebanks seem to be related to the traditional terminology and concepts used in the description of individual languages. These are presumably the result of conversion schemes used when transforming different tagsets into UD. Especially with smaller treebanks improvements could be made relatively fast. However, since the inconsistency is large, it may not always be evident what the best shared solution is. The phenomena pointed out in this paper could be taken into account when systematizing the Uralic treebanks in future releases, although some of the work certainly falls beyond this language family into wider questions around cross-linguistic comparability in Universal Dependencies treebanks. One solution could be to create an explicit mapping between grammatically similar phenomena in the treebanks, and provide harmonization scripts that would adjust different phenomena into comparable representations. This could be connected to better documentation of the treebank conventions, ideally in a machine readable format, so that similar phenomena in different languages could be automatically linked to one another.

6 Acknowledgements

We want to thank three anonymous reviewers for their useful comments. Niko Partanen's work has been carried out within the project Language Documentation meets Language Technology: the Next Step in the Description of Komi, funded by the Kone Foundation.

References

G. F. Fediunova. 2000. *Önija komi kyv, Morfologia*. Komi nebög ledzanin (Коми небӧг лэдзанін). – Syktyvkar. ISBN 5-7555-0689-2.

Ciprian Gerstenberger, Niko Partanen, Michael Rießler, and Joshua Wilbur. 2016. Utilizing language technology in the documentation of endangered Uralic languages. 4:29–47.

Mika Hämäläinen and Jack Rueter. 2018. Advances in Synchronized XML-MediaWiki Dictionary Development in the Context of Endangered Uralic Languages. In *Proceedings of the Eighteenth EURALEX International Congress*, pages 967–978.

Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. 2014. Building the essential resources for Finnish: the Turku Dependency Treebank. *Language Resources and Evaluation*, 48(3):493–531.

Csilla Horváth, Norbert Szilágyi, Veronika Vincze, and Ágoston Nagy. 2017. Language technology resources and tools for Mansi: an overview. In *Proceedings of the Third Workshop on Computational Linguistics for Uralic Languages (IWCLUL 2017)*, pages 56–65.

Mika Hämäläinen. 2019. UralicNLP: An NLP library for Uralic languages. *Journal of Open Source Software*, 4(37):1345.

Boglárka Janurik. 2015. The emergence of gender agreement in code-switched verbal constructions in Erzya-Russian bilingual discourse. *Language Empires in Comparative Perspective. De Gruyter*, pages 199–218.

Marja Leinonen. 2000. Evidentiality in Komi Zyryan. In Lars Johanson and Bo Utas, editors, *Evidentials: Turkic, Iranian and neighbouring languages*, pages 419–440. Walter de Gruyter.

Sjur Moshagen, Jack Rueter, Tommi Pirinen, Trond Trosterud, and Francis M. Tyers. 2014. Open-Source Infrastructures for Collaborative Work on Under-Resourced Languages. In *The LREC 2014 Workshop "CCURL 2014 - Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era"*, pages 71–77.

Kadri Muischnek, Kaili Müürisep, Tiina Puolakainen, Eleri Aedmaa, Riin Kirt, and Dage Särg. 2014. Estonian dependency treebank and its annotation scheme. In *Proceedings of 13th Workshop on Treebanks and Linguistic Theories (TLT13)*, pages 285–291.

Kadri Muischnek, Kaili Müürisep, and Tiina Puolakainen. 2016. Estonian Dependency Treebank: from Constraint Grammar tagset to Universal Dependencies. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Gabrielė Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Andre Kaasen, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Arne Köhn, Kamil Kopacewicz, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Yuan Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Ro□ca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Abigail Walsh Sarah McGuinness, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Takaaki Tanaka, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2019. Universal dependencies 2.4. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Niko Partanen, Rogier Blokland, KyungTae Lim, Thierry Poibeau, and Michael Rießler. 2018. The first Komi-Zyrian Universal Dependencies Treebanks. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 126–132.

Tommi Pirinen. 2019. Building minority dependency treebanks, dictionaries and computational grammars at the same time – an experiment in Karelian treebanking. In *Proceedings of the Third Workshop on Universal Dependencies (UDW 2019)*.

Sampo Pyysalo, Jenna Kanerva, Anna Missilä, Veronika Laippala, and Filip Ginter. 2015. Universal dependencies for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (Nodalida 2015)*, pages 163–172.

Jack Rueter and Francis Tyers. 2018. Towards an open-source universal-dependency treebank for Erzya. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages (IWCLUL 2018*, pages 106–118.

Mariya Sheyanova and Francis M. Tyers. 2017. Annotation schemes in North Sámi dependency parsing. In *Proceedings of the 3rd International Workshop for Computational Linguistics of Uralic Languages (IWCLUL 2017)*, pages 66–75.

Francis M Tyers and Tommi A Pirinen. 2016. Intermediate representation in rule-based machine translation for the Uralic languages. In *Proceedings of the Second International Workshop for Computational Linguistics of Uralic Languages (IWCLUL 2016)*.

Veronika Vincze, Katalin Simkó, Zsolt Szántó, and Richárd Farkas. 2017. Universal Dependencies and morphology for Hungarian - and on the price of universality. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 356–365, 01.

Çağri Çöltekin. 2016. (When) do we need inflectional groups? In *Proceedings of The First International Conference on Turkic Computational Linguistics*, pages 38–43.

ConlluEditor: a fully graphical editor for Universal dependencies treebank files

Johannes Heinecke
Orange Labs
2 rue Pierre Marzin
F - 22307 Lannion cedex
johannes.heinecke@orange.com

Abstract

In this paper we present the ConlluEditor annotation tool for manual annotation of files in CoNLL-U format, such as Universal Dependencies treebanks. Apart from providing a graphical editor for basic and enhanced dependencies, multi-token words, it also runs validation scripts to find potential errors. ConlluEditor uses a client-server architecture. It is freely-available under the 3-Clause BSD License.

1 Introduction

The Universal dependencies (UD) project (Nivre et al., 2016) currently contains more than 140 treebanks in nearly 80 languages, all annotated following the same guidelines for POS tags (UPOS) and dependency relations and stored in a standard format, CoNLL-U. Some treebanks have been converted from an original annotation scheme, whereas others have been annotated from scratch using the UD annotation guidelines. Annotating new sentences for a treebank with a text editor is diffcult, that is why many editing tools exist to help annotators.

2 Related Work

The UD project lists some of the most prominent editors for UD data[1]. We describe a subset of these editors, which we used for various annotation tasks, in order to explain the motivation for this work. We do not go into detail for annotation tools, which do not handle (import/export) CoNLL-U format.

BRAT (Stenetorp et al., 2012) is a powerful annotation tool, which can be configured in detail for manu different annotation tasks, like named entities, coreferences, POS tagging, and dependency relations. BRAT uses an internal character-based format, which can be transformed into CoNLL-U. Since the annotation is character-based, it is the annotator who can decide the start and the end of the tokens. BRAT uses a server and a frontend within a web browser. The display of the annotations, however, becomes confusing if the linked tokens are too wide apart to be shown on a single line.

WebAnno (Eckart de Castilho et al., 2016) is like BRAT a general annotation tool, as powerful as the former and highly configurable through a web interface. As BRAT WebAnno is multi-user and annotations can be made in parallel. apart from UD's CoNLL-U format, WebAnno reads and writes many other standard annotation formats, depending on the annotation tasks. As BRAT, WebAnno displays dependency trees in a flat graph mode (cf. figure 3). Long sentences are wrapped into multiple line, which makes it sometimes difficult to grasp complex dependencies.

UD Annotatrix (Tyers et al., 2018) is a lightweight browser based annotation tool for UD treebanks. Dependency trees can either be edited in a graphical (flat graph) mode or by directly modifying the underlying CoNLL-U data. Annotatrix provides tools to modify the tokenisation (splitting or joining tokens). Apart from CoNLL-U other formats can be used, including plain text. Dependency graphs can be exported as image or LaTeX-code.

Arborator (Gerdes, 2013) permits, like WebAnno, a collaborative annotation of dependency corpora. As Annotatrix, the dependency graphs can be edited in a graphical mode or by modifying the underlying

[1] https://universaldependencies.org/tools.html

CoNLL-U code. Like WebAnno, the corpus being annotated is curated by an administrator. Several annotations of a single sentence can be compared to find the best annotation. Arborator provides a complex search language which enables the annotator to find examples of existing annotations. The dependency graphs can be exported in several image formats. Both Arborator and Annotatrix display sentences as graphics, without wrapping into multiple lines.

3 Motivation

Why yet another tool? Every tool has advantages and inconveniences, either technical issues (online/offline, needs a server or not, provides a functionality like searching or validating or not) or ergonomic ones (size of graphs/trees displayed, edition mode).

When we started working with UD treebanks, the then existing tools to graphically display dependency trees did not provide all display and search functions we needed. Further, for a demo we needed an offline solution. Finally, from a personal point of view, flat dependency graphs are more difficult to understand than dependency trees, which show horizontally the dependents. Notably in long sentence, one can see quickly the clausal structure. What started as a quick javascript hack to display dependency trees graphically, grew finally into a fully fledged editor which we tested in correcting existing CoNLL-U files and annotations of new sentences from scratch. Annotating dependency relations and correcting token information like form, lemma UPOS, XPOS and features proved to be really fast with this tool.

ConlluEditor provides the following features, which will be explained in the remaining sections:

- full graphical editor for basic and enhanced dependency relations
- word edit (form, lemma, UPOS, XPOS, features, misc-column)
- autocompletion (UPOS, XPOS, deprels; lists of valid labels must be provided)
- editing multitoken words (`[1-2] ...`)
- support for empty nodes (`[5.1] ...`)
- comment editing
- support for right-to-left scripts like Arabic or Hebrew
- split and join words (to correct bad tokenization)
- split and join sentences (to modify sentence segmentation)
- undo/redo
- regex search functions (including sequences of tokens, sub-graphs and comments)
- git support (add/commit)
- validation: indicates undefined UPOS, XPOS, dependency relations (based on lists given to the server)
- prohibition of invalid (cyclic) trees
- normalisation of token ids (first column, from 1 to *n*, taking into account multitoken words, empty words and heads)
- validation with external script (such like UD's `validate.py`[2]) on the current sentence
- export of dependency graphs as `.svg` image, LaTeX-code (for the tikz-dependenciy[3] package or the `deptree.sty`[4] LaTeX style, provided by ConlluEditor), sd-parse[5], CoNLL-U
- limited multi-user support: as long as two users do not edit the same sentence

ConlluEditor does not (yet) allow collaborative working like Arborator or WebAnno or other tools. So each annotator works on an individual copy which have to be merged in a second step.

The ConlluEditor is under the 3-Clause BSD License[6] and available at `https://github.com/Orange-OpenSource/conllueditor`.

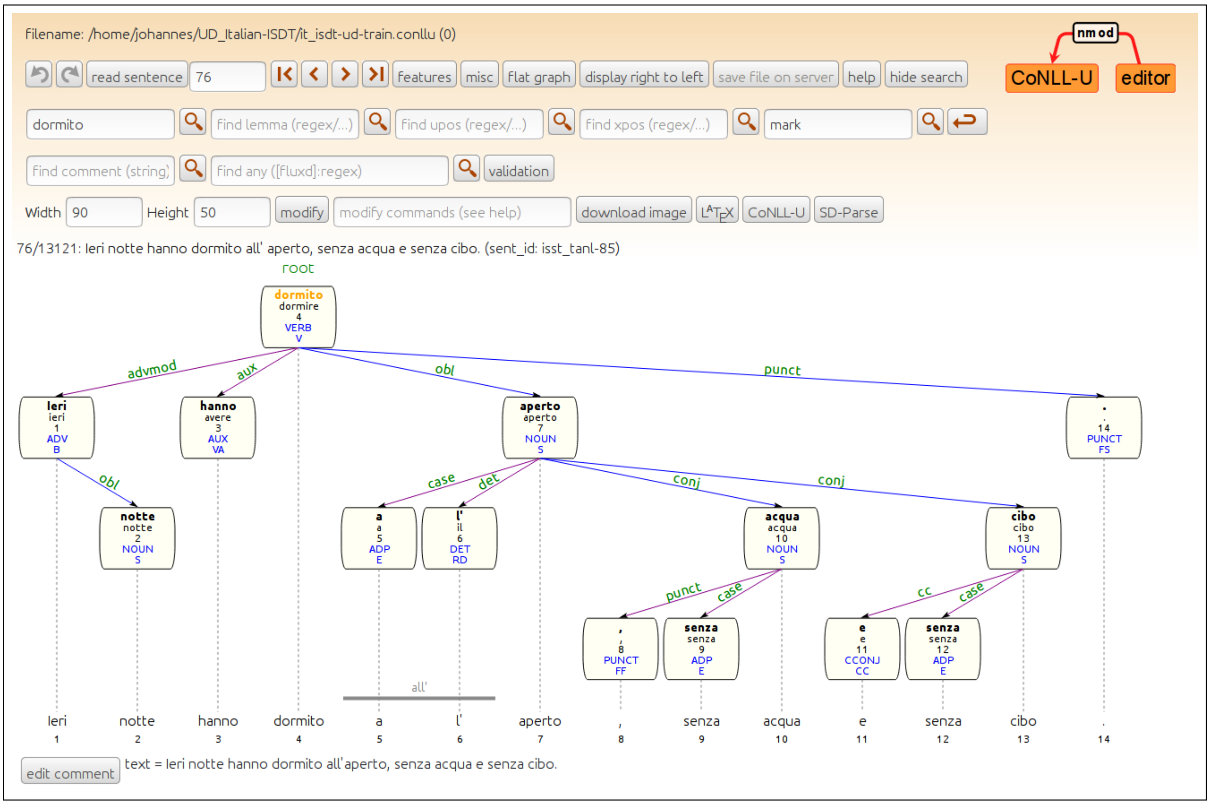
[2] https://github.com/UniversalDependencies/tools.git

[3] https://ctan.org/pkg/tikz-dependency

[4] https://github.com/Orange-OpenSource/conllueditor/blob/master/doc/deptree-doc.pdf

[5] http://nlp.stanford.edu/software/stanford-dependencies.shtml

[6] https://opensource.org/licenses/BSD-3-Clause



Filename: /home/johannes/UD_Italian-ISDT/it_isdt-ud-train.conllu (0)

read sentence 76

dormito

mark

Width 90 Height 50

76/13121: Ieri notte hanno dormito all' aperto, senza acqua e senza cibo. (sent_id: isst_tanl-85)

text = Ieri notte hanno dormito all'aperto, senza acqua e senza cibo.

Figure 1: Main view in tree mode (note the multitoken word at nodes 9 and 10). The search buttons can be hidden to have place for deeper trees. Morpho-syntactic features can be displayed or hidden (to gain vertical space). Search results are highlighted (here *dormito*). The horizontal and vertical spacing can be modified to accomodate longer sentences. In order to change the font size, a `.css` style sheet can be modified. A help-Button provides detailed help on how to use the editor interface. (Example taken from the UD Italian-ISDT treebank).

4 Features

ConlluEditor is a server/client architecture with a server implemented in java which reads and manages the CoNLL-U file, writes modifications to disk and commits the changes to a git repository (if under git control). The front-end is implemented in javascript (using Bootstrap and jQuery), the main edit view is shown in figure 1. The data is passed between client and server using AJAX. The front-end has been tested with recent versions for Firefox, Chrome and Edge on Linux and Windows 10.

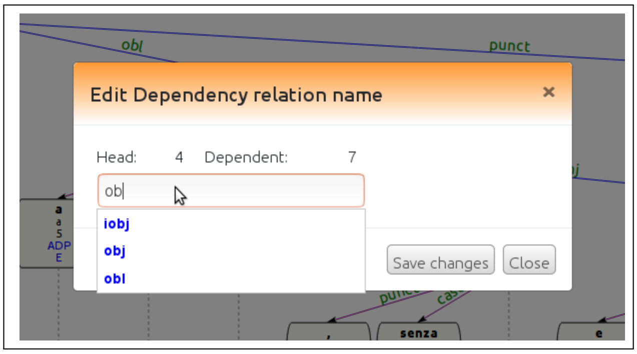
4.1 Graphical editing

Annotating dependency relations graphically speeds up the annotation work enormously. In ConlluEditor a simple click on a word (the future dependent) and a subsequent click on a second word (head) establishes a dependency relation, a dependency label edit window (fig. 2) opens to set the dependency relation label. Clicking twice on the same word makes this word root.

The graphical display of dependency trees is often a matter of personal taste. For space reasons, dependency graphs are often presented as flat graphs (e.g. generated with the tikz-dependencies package for LaTeX). In order to visually understand the syntax of a tree, a tree presentation (cf. figure 1) is often clearer. ConlluEditor proposes both views (check-button). Figure 3 shows the flat mode (here, the search functions are hidden).

In order to modify the word, a double-click or Control-click opens the word edit window (fig. 4). In order to avoid errors on UPOS, XPOS or dependency labels, ConlluEditor reads lists of valid labels and provides autocompletion. It also highlights invalid values in the main view.

Multitoken words can be edited (first and last word, multitoken wordform), by clicking on the multi-

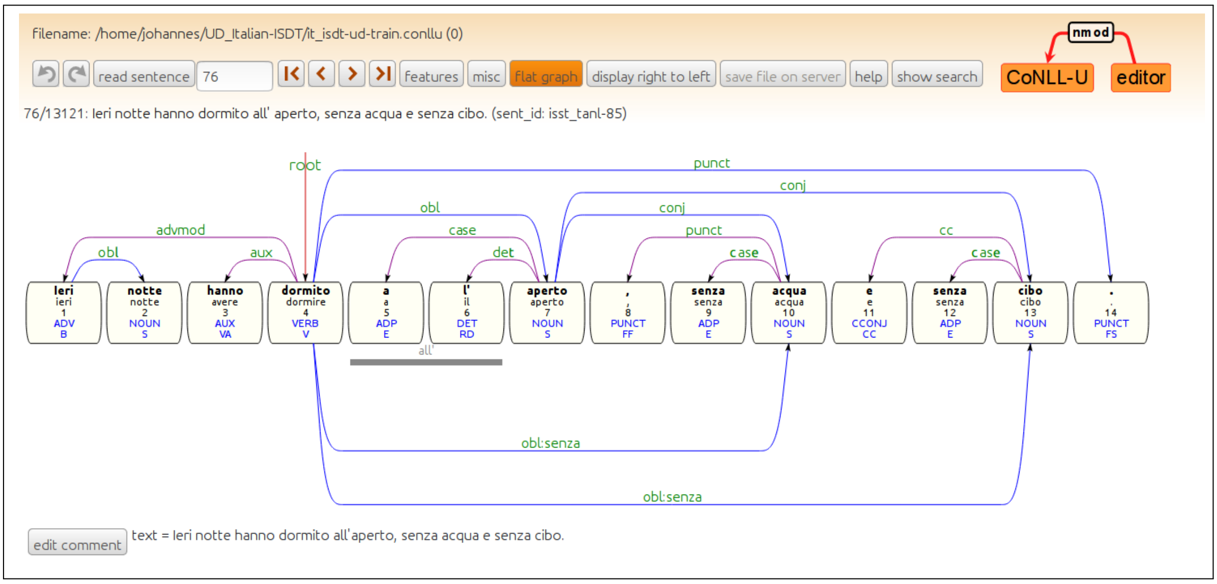

Edit Dependency relation name

Head: 4 Dependent: 7

obl

Figure 2: Dependency label edit window. All possible labels are proposed via autocompletion. Invalid labels, however, are accepted.

token indicator.


Filename: /home/johannes/UD_Italian-ISDT/it_isdt-ud-train.conllu (0)

76/13121: Ieri notte hanno dormito all' aperto, senza acqua e senza cibo. (sent_id: isst_tanl-85)

text = Ieri notte hanno dormito all'aperto, senza acqua e senza cibo.

Figure 3: Dependency tree in flat mode. Multitoken words are marked as in tree mode. In flat mode, enhanced dependencies are also displayed and can be edited.

4.2 Searching

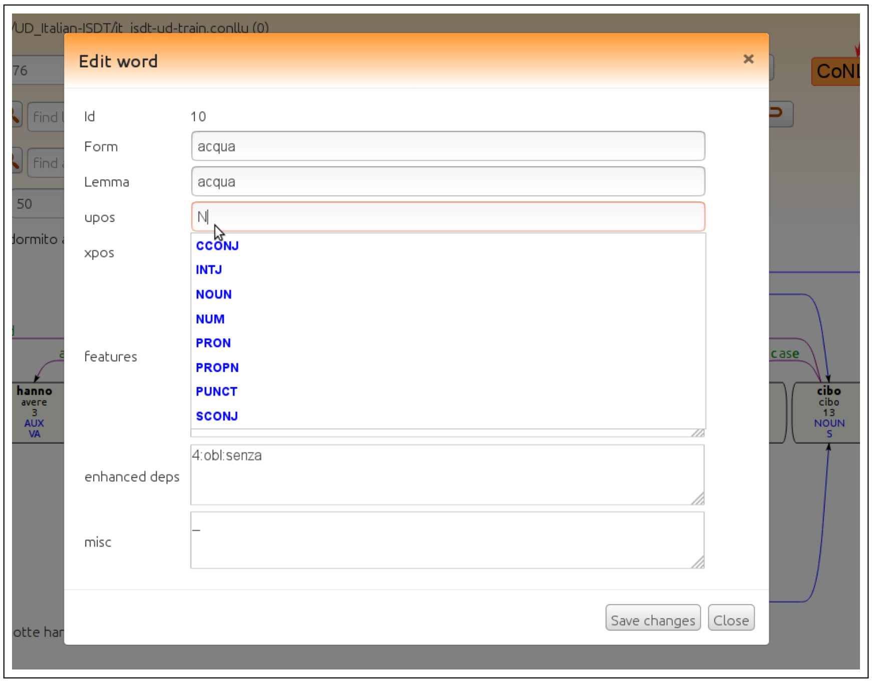
Once treebanks grow in size, search for similar phrases becomes extremely useful, not only to annotate similar structures in an similar way. ConlluEditor provides searching (with regular expressions) for forms, lemmas, UPOS, XPOS, dependency relations and comments. Searching sequences of forms, lemmas etc. is also possible, as well as a combination of these, like searching for all UPOS `PRON` which precede the lemma *fish*. It is, however, much less powerful than, for example, the Grew system (Bonfante et al., 2018)[7], which provides a complex query language to find sub-trees/sequences of forms, lemmas, UPOS, etc.

4.3 Data export and *git* support

Currently ConlluEditor reads and saves only files in the CoNLL-U standard[8]. However the interface permits to get the current sentence either as an `.svg` image or as code for LaTeX (tikz-dependencies), sd-parse or CoNLL-U. A simple click on the corresponding button opens a window to copy the generated code. The LaTeX-code includes enhanced dependencies.

[7] http://match.grew.fr/

[8] http://universaldependencies.org/format.html



Edit word

Id 10

Form acqua

Lemma acqua

upos N

CCONJ
INTJ
NOUN
NUM
PRON
PROPN
PUNCT
SCONJ

xpos

features

enhanced deps 4:obl:senza

misc _

Save changes Close

Figure 4: Word edit window (proposed a second method of editing enhanced dependencies). For UPOS, XPOS and dependency relation labels, the editor proposes autocompletion of valid values.

If the edited CoNLL-U file is under git version-control, ConlluEditor performs a *git add* and *git commit* on every edit. The commit message contains information about the document, and the modified word and sentence. In order to have less commits, a server option permits to have commits only after *n* edits.

4.4 Display modes

Apart from the two layout modes (tree and flat), ConlluEditor supports languages written in right-to-left mode like Arabic or Hebrew (figure 5).

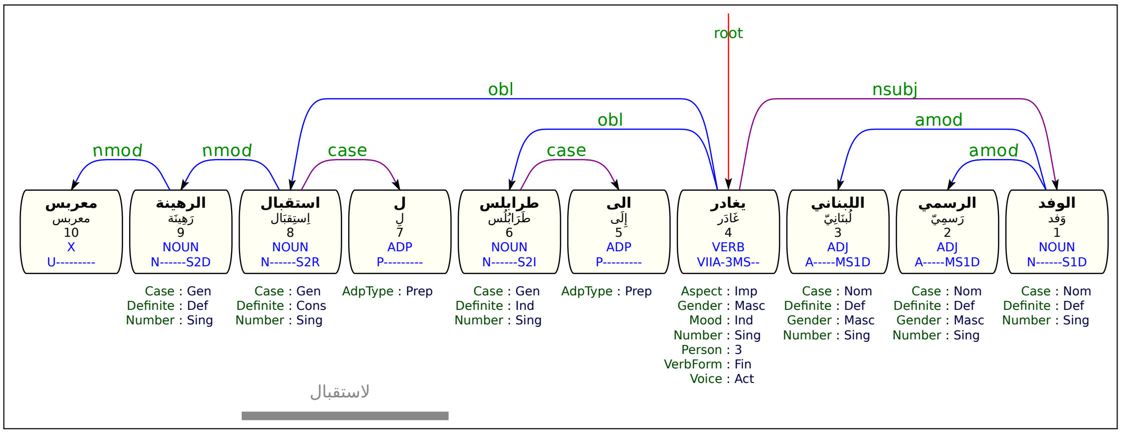


Figure 5: Support for languages which write right-to-left (example taken from the UD Arabic-PADT treebank). Here the display of morphological features is activated.

Whereas enhanced dependencies are currently only shown in flat mode, empty nodes (using the `[5.1]` format in the ID column) are shown in both modes (cf. fig. 6).

4.5 Validation of annotated sentence

ConlluEditor can run any validation programme on the current sentence. A validation script and its arguments must be defined in a configuration script which is given to the ConlluEditor server. Clicking

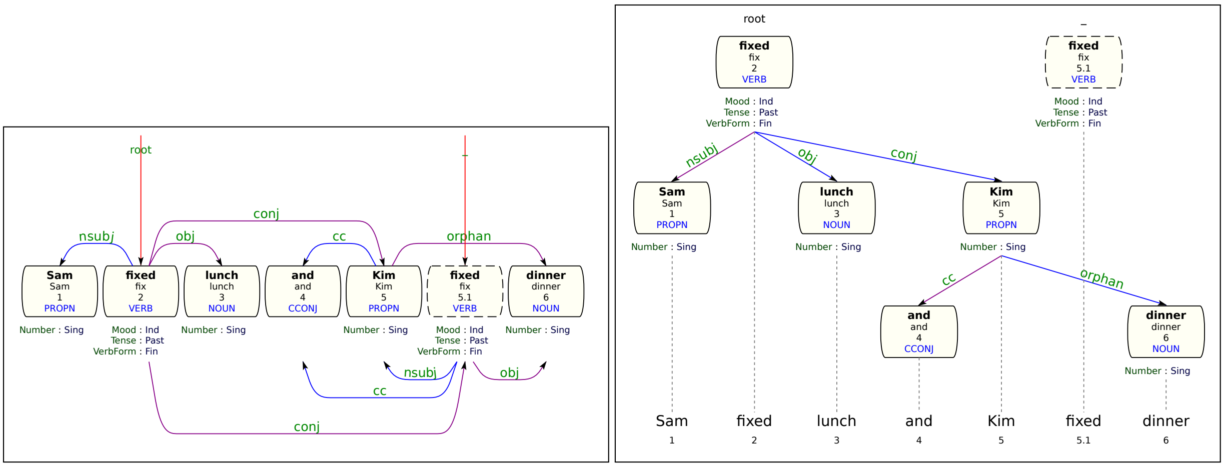
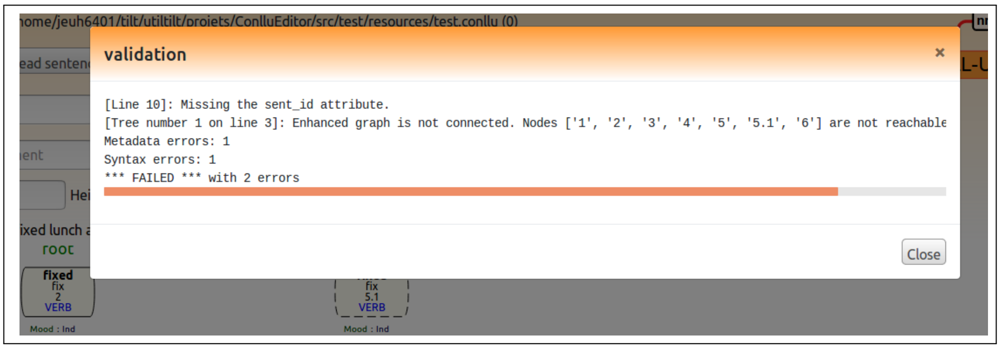


Figure 6: Display of empty nodes in both modes (also showing morphological features)

on the validation button runs the script on the current sentence and shows the entire output in a window (fig. 7).


home/jeuh6401/tilt/utiltilt/projets/ConlluEditor/src/test/resources/test.conllu (0)
validation
×
Close

```
[Line 10]: Missing the sent_id attribute.
[Tree number 1 on line 3]: Enhanced graph is not connected. Nodes ['1', '2', '3', '4', '5', '5.1', '6'] are not reachable
Metadata errors: 1
Syntax errors: 1
*** FAILED *** with 2 errors
```

Figure 7: Output of the validation programme

5 Future work

The main features we plan to implement is support for CoNLL-U Plus (`.conllp`) files[9] and finally add (language specific) validation rules, to find formal errors or inconsistencies, cf. also Marneffe et al. (2017). Even though CoNLL-U has become a standard for syntax annotations, other formats exists. ConlluEditor currently exports three formats (`conllu`, `sd-parse` and `latex`), others can be implemented rapidly if needed.

References

Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*. Wiley-Iste.

Richard Eckart de Castilho, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures. In *Proceedings of the International Conference on Computational Linguistics COLING*, pages 76–84.

Kim Gerdes. 2013. Collaborative Dependency Annotation. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing)*, pages 88–97, Prag.

[9] http://universaldependencies.org/ext-format.html

Marie-Catherine de Marneffe, Matias Grioni, Jenna Kanerva, and Filip Ginter. 2017. Assessing the Annotation Consistency of the Universal Dependencies Corpora. In *Proceedings of the Fourth International Conference on Dependency Linguistics*, pages 108–115, Pisa, Italy.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Yoav Goldberg, Jan Hajič, Manning Christopher D., Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *the tenth international conference on Language Resources and Evaluation*, pages 23–38, Portorož, Slovenia. European Language Resources Association.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a Web-based Tool for NLP-Assisted Text Annotation. In *13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Francis M. Tyers Tyers, Mariya Sheyanova, and Jonathan North Washington. 2018. UD Annotatrix: An annotation tool for Universal Dependencies. In *ACL 2018*, pages 10–17, Melbourne.

Towards an adequate account of parataxis in Universal Dependencies

Lars Ahrenberg
Department of Computer and Information Science
Linköping University
lars.ahrenberg@liu.se

Abstract

The parataxis relation as defined for Universal Dependencies 2.0 is general and, for this reason, sometimes hard to distinguish from competing analyses, such as coordination, conj, or apposition, appos. The specific subtypes that are listed for parataxis are also quite different in character. In this study we first show that the actual practice by UD-annotators is varied, using the parallel UD (PUD-) treebanks as data. We then review the current definitions and guidelines and suggest improvements.

1 Introduction

The aims of the Universal Dependencies (UD) project are high and somewhat conflicting, (cf. Osborne and Gerdes, 2019). While the emphasis is on linguistic typology and parsing, there are no restrictions on the kind of linguistic data that UD could be applied to. Indeed, the UD treebanks cover as varied genres as tweets and literature. Adding to this the desired goal that UD parsers should have high accuracy on text which has not been tokenized, we have a situation where UD has to deal not just with common syntactic constructions, but with a multitude of genre-specific and stylistic varieties.

As an annotator one frequently encounters cases where the choice between *parataxis* and some other relation is unclear. Take (1) as an example:[1]

(1) A very good performer she was; breaking into Yiddish, into Italian, into German, accenting and gesturing, turning now into a claque of elderly Jews, now into a frightened small boy.

We have one typical instance of parataxis, supported by the semi-colon and relating the head of what follows it, *breaking*, to the head of the clause preceding it, *performer*. But what about the verbs *accenting* and *turning*? Are they conjuncts to *breaking* or related via parataxis or maybe adverbials? The sequence of prepositional phrases, *into Yiddish, into Italian, into German* could similarly be seen as coordinated with *Yiddish* as the head, related via parataxis, or even be analysed as independent oblique constituents in relation to *breaking*. The last part of the sentence forms what Matthews (**?**) calls a correlative construction; two clauses or phrases held together by occurrences of a word pair, here *now – now*, the relation of which is open to interpretation.

Part of the problem for a UD annotator here is that UD does not require a conjunction to be present with a coordinated conjunct. As a matter fact, the classical quote attributed to Caesar, *Veni, vidi, vici*, is analysed as a coordination in the UD guidelines[2], while in many works in stylistics, it tends to be described as a prime example of parataxis[3]. Another factor is that UD so far has not been specific enough about the categories that can enter into a parataxis relation.

The purpose of this paper is twofold. First, we study how the parataxis relations and its competitors have been used in the Parallel UD (PUD) Treebanks. The choice of these treebanks is motivated by the fact that the sentences in them are close translations of one another, mostly from English source

[1] All examples that are not from the PUD treebanks are taken from, or modelled upon, sentences found in (**?**)

[2] https://universaldependencies.org/u/dep/conj.html

[3] See, for example, https://literarydevices.net/parataxis/

sentences. Thus, their annotations should be as similar as can be found in the full set of UD treebanks. Second, in view of problems we have encountered in the annotation of paratactic style in literature, we review the current guidelines for paratactic relations, in particular *parataxis*, with the goal of suggesting improvements and clarifications.

Paratactic relations are of interest both to grammar and stylistics. Matthews (**?**) argues that there is no clear-cut border, and this is, we believe, partly what causes the difficulty for UD annotators. Halliday (**?**) provides a detailed analysis of the opposition hypotaxis-parataxis within his broader systemic model that also involves 'logico-functional' aspects of relations between syntactic units. This level is however not available in UD representations.

2 Paratactic relations in UD

The UD framework offers a limited number of paratactic relations, or loose joining relations as they are called in (de Marneffe et al., 2014). The most common are *conj* that covers all forms of coordination, *parataxis* used for (mostly) asyndetic sequences of clauses, and *appos* for co-referring nominals that come one after the other, with or without a comma in between. Other paratactical relations are *list*, used for sequences of small information units, *flat* used for multi-token names, *fixed* for fixed lexical items, and *discourse* for interjectional items.

The relations *list*, *parataxis*, and *appos* are needed as they give 'a robust analysis of more informal forms of text (de Marneffe et al., 2013; de Marneffe et al., 2014). Parataxis is said to be needed also with more formal writing for constructions such as sentences joined with a colon. In (de Marneffe et al., 2013) we get an example of the use of list: *Steve Jones Phone:555-9814 Email:jones@abc.edf* where *Phone* and *Email* depends on *Jones* via *list*. This example is found also in the guidelines, though *Steve* is now the head node.

In the current guidelines, *parataxis* is described as 'a relation between a word (often the main predicate of a sentence) and other elements, such as a sentential parenthetical or a clause after a ':' or a ';', placed side by side without any explicit coordination, subordination, or argument relation with the head word. The guidelines go on to list five sub-types: side-by-side sentences, reported speech, interjected clauses, tag questions, and news article bylines.

The guidelines for other languages follow the ones for English to a large extent. Some languages, including Italian (**?**) and French (Gerdes and Kahane, 2017), have developed language-specific extensions for parataxis.

Apart from the other paratactic relations, we find several hypotactic relations, and especially *ccomp*, that competes with *parataxis* for the analysis of certain constructions.

3 Parataxis in the PUD treebanks

Most of the parallel UD treebanks were developed for the CoNLL 2017 shared task on Multilingual Parsing from Raw Text to Universal Dependencies (**?**). Others have been developed afterwords, such as those for Czech, Finnish, and Swedish. All PUD treebanks contain 1000 sentences of which the first 750 are originally English, while the remaining are originally German, French, Italian, or Spanish.

Translations to other languages were made via English. For this reason, we take the English PUD treebank as our reference. It has 97 instances of parataxis, the majority of which are either side-by-side sentences or reported speech. The rest we have divided into Parentheticals and Other (see Table 1). Parenthetical covers interjected clauses and news article bylines, but the majority are inserted or added

Subtype	**Frequency**
Side-by-side sentences	44
Reported speech	42
Parentheticals	8
Tag questions	0
Other	3
Total	97

Table 1: The parataxis relation in English PUD distributed on subtypes.

comments of the kind illustrated in (2).[4] The Other class is used for a few cases which we believe are better analysed as something else. An example is (3).

(2) (a) Their first king was **Mojmír** I (**ruled** 830–846).
(w01010047) *parataxis(Mojmir,ruled)*
(b) And, she granted, "you have to look at where she has acknowledged that we **need** to do something different–we can **do** better–and where she has expressed regret."
(n01060069) *parataxis(need,do)*

(3) Where does all her energy **come** from? Or that **voice**, which can blast out with a force to induce shockwaves? (n01116018) *parataxis(come,voice)*

The translators were asked to produce as close translations as possible (**?**). Thus, the translations can be expected to follow the structure of the source sentences. This is largely the case, but some languages, such as German, have more constructional differences than others with English. Table 2 shows absolute frequencies for the parataxis relation in the 14 PUD treebanks that have more than 10 instances of it. The variation in numbers is striking given that content and structure are supposedly very similar. The differences become even more pronunced when we look at the distribution of the parataxis relation in the treebanks. Table 3 shows the number of overlapping and non-overlapping relations for English compared to the other languages.

Treebank	**en**	**ar**	**cs**	**de**	**es**	**fi**	**fr**	**hi**	**id**	**it**	**pt**	**ru**	**sv**	**tu**
Frequency	97	24	20	68	106	108	106	93	116	99	103	195	134	74

Table 2: Absolute frequencies for the *parataxis* relation in fourteen PUD treebanks: English, Arabic, Czech, German, Spanish, Finnish, French, Hindi, Indonesian, Italian, Portuguese, Russian, Swedish, and Turkish.

Treebank	**Overlaps**	**English only**	**Other only**	**Similarity**
UD_Arabic-PUD	15	82	9	0.24
UD_Czech-PUD	10	87	10	0.17
UD_Finnish-PUD	73	24	35	0.71
UD_French-PUD	62	35	42	0.62
UD_German-PUD	43	54	25	0.53
UD_Hindi-PUD	40	57	53	0.42
UD_Indonesian-PUD	44	53	70	0.43
UD_Italian-PUD	73	24	26	0.75
UD_Portuguese-PUD	79	18	24	0.79
UD_Russian-PUD	74	23	112	0.52
UD_Spanish-PUD	72	25	34	0.71
UD_Swedish-PUD	86	11	48	0.75
UD_Turkish-PUD	43	54	26	0.52

Table 3: How English instances of *parataxis* are distributed compared to instances in other PUD treebanks. Overlaps have been assumed whenever an English tree and the corresponding tree for the other language have the same number of instances. Similarity is measured as $2 * Overlap/(English + Other + 2 * Overlap)$

Using a simple similarity metric, we can see that the Portuguese treebank is the most similar in its annotation of the parataxis relation. Still, even the English-Portuguese pair has more than 40 cases of non-overlaps. The Swedish treebank has transfered the highest share of English instances (86 out of 97) but has added many extra ones where the English PUD uses hypotactic relations.

A part of the variation can be explained by constructional changes made in the translations, but the found variation is much to great to suggest that the goal of "consistent annotation of similar constructions across languages"[5] has been met.

[4] The label enclosed in parenthesis is the sentence identifier, sent_id, from the PUD files.

[5] https://universaldependencies.org/introduction.html

4 Problematic constructions

A closer look at the PUD data reveals a number of typical cases, where the annotation between languages differ, although the constructions are similar.

4.1 Side-by-side clauses

Asyndetic side-by-side clauses can be annotated either as *parataxis* or *conj*. Both interpretations accord with the guidelines (cf. the Latin example in the introduction). As the guidelines don't provide distinctive information, the decision is up to the annotator's judgement. The following is an English-Italian pair.

(4) EN: I'm **going** to jail either way, **hope** it was worth it
(n01011017) *parataxis(going,hope)*
IT: In entrambi i casi **finirò** in prigione, **spero** ne sia valsa la pena
(n01011017) *conj(finirò,spero)*

German, Swedish and Spanish have also opted for parataxis, whereas French follows Italian, using conj. It should be noted that the presence of a coordinating conjunction is not always disambiguating for annotators. Sentence (3) above is a telling example.

4.2 Reported speech

The guidelines for reported speech clearly distinguish clausal complements from paratactical direct speech. Basically, the difference is that when the reported speech is, or could be, introduced by a subjunction such as *that*, the analysis should be *ccomp*, otherwise it should be *parataxis*. In particular, if there is a clear signal of separation, such as a colon or comma and/or citation marks, *parataxis* should be used.

Many annotators, though, prefer to see the speech verb as governor and the reported speech as complement. This is implemented throughout in the German PUD treebank. Here both the relation and the direction of the relation changes. You may also find examples where the parataxis relation is used, but reversed, as in the French version of (5).

(5) EN: "This is a **disaster** for pain patients," Mailis **said** in an interview ...
(n01041006) *parataxis(disaster,said)*
DE: Das ist eine **Katastrophe** für Schmerzpatienten, **sagte** Mailis in einem Interview ...
(n01041006) *ccomp(sagte,Katastrophe)*
FR: C'est un **désastre** pour les malades en souffrance, a **déclaré** Mailis dans un entretien ...
(n01041006) *parataxis(déclaré,désastre)*

4.3 Parentheticals

In the category of Parentheticals we find cases where parataxis alternates with appos. In the following example, the inserted unit is a noun phrase that may be interpreted as an elliptical clause. English PUD annotates this as parataxis, while Spanish (and German, Swedish) treats it as apposition. The Spanish translation has placed the verb translating *sent* after the phrase referring to dinosaurs. This may explain the actual annotation, but should not really affect it. In French, the translation has inserted the conjunction *et*, suggesting an analysis as conjunct. There is no conjunction in the English and Spanish sentences, but it may nevertheless be inserted (or 'heard') without changing the meaning much.

(6) EN: ... and sent so many **species** - not just the **dinosaurs** - into oblivion .
(n01023034) *parataxis(species,dinosaurs)*
ES: ... y que hizo que muchas **especies**, no solo los **dinosaurios**, cayeran en el olvido.
(n01023034) *appos(species,dinosaurs)*
FR: ... qui ont causé l'extinction de nombreuses **espèces**, et pas uniquement des **dinosaures**.
(n01023034) *conj(espèces,dinosaures)*

4.4 Hypotactic competitors

Apart from the differences in the analysis of reported speech, there are other cases where a parataxis relation in one language corresponds to a hypotactic dependency in another language without any apparent change of structure. The Swedish PUD treebank has several examples of parataxis, where English and other languages prefer *obl*, as in (7). A possible reason for this analysis is the presence of a comma, indicating a pause, before the prepositional phrase.

(7) EN: The issuing of coinage is predominantly **numismatic** in nature, with the **intention** of being sold mainly to collectors. (w04003054) *obl(numismatic,intention)*
SV: Utfärdandet av mynt är företrädesvis **numismatiskt** till sin natur, med **avsikten** att säljas främst till samlare. (w04003054) *parataxis(numismatiskt,avsikten)*

5 Discussion

What is to be done? We could put some of the blame on annotators who don't follow the guidelines properly. But annotators have intuitions, often based in a linguistic tradition, and may violate the guidelines for good reasons.

We could also blame the guidelines for not being complete or clear enough. This is partly true but the guidelines for reported speech are very clear, providing ample examples to clarify the contrast of parataxis to ccomp. But the frequency of ccomp in the analysis of reported speech raises the question why this should be a case of parataxis in the first place. Reported speech is the only type of parataxis where one of the two units is a semantic argument of the other and thus it is motivated to treat it as a case of complementation. The distinction between direct and indirect speech can be done as a specialization of ccomp, if needed.

An even stronger recommendation would be to generally prefer hypotactic relations over paratactic ones in UD, when there are arguments for both alternatives. This would make it clear that a sentence such as (7) has no parataxis relation.

Conversely, we note that UD has authorities such as Halliday (**?**) behind it in treating reported speech as a case of parataxis. It can also be argued that such a move would make the analysis of sentences such as (8) more complicated, as the clausal argument of the speech verb would then be split causing the relation between the two parts to cross the root node. However, non-projectivity cannot be entirely avoided with the current recommendation either and, actually the tree also relates 'leave' to 'time' as the head of a relative clause. The interposing of units is quite common in literary genres and, as they are easy to detect, could actually be given a dependency relation of their own, which as other relations may be further subtyped if needed. To compensate for this addition to the framework, the relation *list* could be deprecated as the kind of constructions it has been used for could equally well be modeled with parataxis.

(8) There **was** a time, Mr Panvalkar **said**, when he felt that they should leave the building.
(n01010042) *parataxis(was, said)*

Turning to the side-by-side sentences, there are two oppositions where annotations typically differ: parataxis vs. conjunct, and parataxis vs. apposition. For the first pair one could take the view that the presence of a coordinating conjunction somewhere in the sequence of units should always result in a conjunct analysis. This is a formalistic approach, but the alternative leaves much to annotators' varying intuitions. Moreover, when no coordinating conjunction is present, we need to know what information can guide an analysis as conjunct. It is possible to test whether a conjunction can be inserted without change of meaning and use this as a guideline. Sometimes, however, the insertion of a conjunction would lead to a substantial change of style, and thus be felt as the use of a different construction. This happens, inter alia, when a unit is repeated or slightly varied for reasons of emphasis or other stylistic effect, as in (9).

(9) (a) It's my mistake, my mistake.
(b) I had to let go of my detachment, my resentment.

Although the two units are filling the same slot in the larger sentential context, they are not joined by a coordinating conjunction, and cannot be without changing the construction's character. Given that the units are noun phrases, and in some sense, co-referring, it is tempting to analyse their relation as appositional. However, clauses can also be similarly repeated or varied, and this may speak in favour for an analysis as parataxis.

Another type repeats a structure, but each unit adds a different aspect, as in (10). In this case it is easy to hear a 'but' before the last unit in the sequence and treat the whole as a sequence of conjuncts. On the other hand, if the author had wanted a conjunction there, she would presumably have put one in. It would be somewhat disrespectful to ignore her choice.

(10) Her waist was curved, her legs were long, her breasts round, her stomach was flat, her bottom was not.

As regards parataxis vs. apposition the guidelines for the appos relation says that it relates a noun (head of a noun phrase) to a nominal, where the latter, dependent part is often optional. If parataxis relates clauses to one another, and appos relate nominals, we must nevertheless reckon with cases where a clausal head has a nominal side-by-side dependent, and this often leads to different analyses, as in (7) above. Moreover, we have sentences where the choice of head is not evident. Witness (11):

(11) EN: Greece was **divided** into many small, self-governing communities, a **pattern** largely dictated by Greek geography: ...
(w03005015) *parataxis(divided, pattern)*
DE: Griechenland war in viele kleine eigenständige Kommunen **unterteilt** - eine **Form**, die weitgehend durch die griechische Geografie vorgegeben ist: ...
(w03005015) *obl(unterteilt, Form)*
FR: Le pays est alors divisé en une **multitude** de petites communautés indépendantes, **situation** imposée par la géographie grecque
(w03005015) *appos(multitude, situation)*

In the French annotation, a noun is selected as head and the relation is taken as an apposition. In German the annotators have opted for an oblique relation which might be an error. Italian, Spanish and Swedish follow the English analysis. In the reverse situation, i.e., with a nominal head and a clausal dependent, the recommended analysis is acl, even for non-restrictive relative clauses.

(12) By comparison, it cost $103.7 million to build the NoMa infill Metro **station**, which **opened** in 2004. (n01005023) *acl:relcl(station,opened)*

However, adverbial and prepositional phrases can also be sequenced and then the distinction between hypotaxis and parataxis gets blurred. Look again at (1), repeated here for convenience as (13):

(13) A very good performer she was; breaking into Yiddish, into Italian, into German, accenting and gesturing, turning now into a claque of elderly Jews, now into a frightened small boy.

The *obl* relation has in principle no limitation on its number of occurrences in a single clause. For this reason one can be motivated to annotate not only *Yiddish* but also *Italian* and *German* as dependent on *breaking* via obl. A similar analysis could be proposed for *boy* with *turning* as its head. However, oblique units that belong to the same slot of the predicate have a stronger affinity than those who do not, which motivates a paratactical analysis. And although there are several places where the conjunction *and* can be inserted, suggesting a conjunct analysis, its absence may be taken as a decisive criterion for using parataxis.

(14) I could see her, hair and flesh escaping, hope trapped inside.

A similar problem arises in (14): should *escaping* be seen as dependent on *see* or *her*? In the first case we can choose between parataxis and advcl (hearing an absent *with* before *hair*), in the second acl would be the most appropriate relation. In a case like this, one cannot escape relying on annotators' individual interpretations, but in many of the other examples we believe a further elaboration of the guidelines could be quite helpful. Suggestions are given in the summary:

- The use of the *parataxis* relation varies quite a bit among UD annotators. This is not only caused by language or data differences, but differences in annotation practices, as evidenced by our study of annotations in the PUD treebanks.
- There is no need for more than one general and broadly defined relation of paratactic sequencing, *parataxis*. This relation may be further subtyped for the genres and treebanks that need it, as done for UD_Italian-Postwita (**?**) and UD_French-Spoken (Gerdes and Kahane, 2017). The *list* relation can be deprecated.
- The guidelines on *parataxis* need to be developed. The account of reported speech sets a good standard in terms of the level of detail, but should be motivated further.
- Just as UD have other basic principles, such as favouring content words and left constituents as heads, a basic principle stating preferences for hypotactic relations over paratactic could be adopted. Exceptions can be allowed, as is currently done for reported speech, and, as suggested here, for asyndetic sequences of units (clauses and phrases) expressing the same semantic role.
- Detailed guidelines regarding the possibility to determine a relation on the basis of inserting words that are not there should be worked out. The simplest guideline would be to completely disallow such argumentation except in cases of ellipsis, when the missing word can be inferred from the immediate context.

References

Marie-Catherine de Marneffe, Miriam Connor, Natalia Silveira, Samuel R. Bowman, Timothy Dozat and Christopher D. Manning. 2013. More constructions, more genres: Extending Stanford dependencies. *Proceedings of the 13th International Conference on Dependency Linguistics.*

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre and Christopher D. Manning. 2014. Universal Stanford Dependencies: A cross-linguistic typology. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14).* 4585–4592.

Kim Gerdes and Sylvain Kahane. 2017. Trois schémas d'annotation syntaxique en dépendance pour un même corpus de français oral: le cas de la macrosyntaxe. *Actes de la 24e conférence sur le traitement automatique des langues (TALN), Atelier sur les corpus annotés du français (ACor4French), Orléans.*

M. A. K. Halliday 1985. *An Introduction to Functional Grammar.* Edward Arnold.

P. H. Matthews 1981. *Syntax.* Cambridge University Press.

Timothy Osborne and Kim Gerdes. 2019. The status of function words in dependency grammar: A critique of Universal Dependencies (UD). *Glossa: a journal of general linguistics* 4(1): 17. 1–28. DOI: https://doi.org/10.5334/gigl.537.

Manuela Sanguinetti, Cristina Bosco, Alberto Lavelli, Alessandro Mazzei, Oronzo Antonelli and Fabio Tamburini. PoSTWITA-UD: an Italian Twitter Treebank in Universal Dependencies. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018).* https://www.aclweb.org/anthology/L18–1279.

Jennette Winterson 1997. *Gut symmetries.* Granta Books, London.

Daniel Zeman, Martin Popel, Milan Straka, ..., and Josie Li 2017. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Vancouver, Canada, 3–4 August 2017.* Association for Computational Linguistics.

Recursive LSTM Tree Representation for Arc-Standard Transition-Based Dependency Parsing

Mohab Elkaref[*]
IBM Research
Daresbury, UK
mohab.elkaref@ibm.com

Bernd Bohnet
Google Inc.
London, UK
bohnetbd@google.com

Abstract

We propose a method to represent dependency trees as dense vectors through the recursive application of Long Short-Term Memory networks to build Recursive LSTM Trees (RLTs). We show that the dense vectors produced by Recursive LSTM Trees replace the need for structural features by using them as feature vectors for a greedy Arc-Standard transition-based dependency parser. We also show that RLTs have the ability to incorporate useful information from the bi-LSTM contextualized representation used by Cross and Huang (2016) and Kiperwasser and Goldberg (2016b). The resulting dense vectors are able to express both structural information relating to the dependency tree, as well as sequential information relating to the position in the sentence. The resulting parser only requires the vector representations of the top two items on the parser stack, which is, to the best of our knowledge, the smallest feature set ever published for Arc-Standard parsers to date, while still managing to achieve competitive results.

1 Introduction

Neural network-based dependency parsers have typically relied on combination of raw features, as represented by their dense vector embeddings to represent features of a sentence as well as the parser state (Chen and Manning, 2014; Weiss et al., 2015; Andor et al., 2016; Zhou et al., 2015). On the other hand, there has been substantial work on using innovative deep learning architectures to build more informative feature representations.

One approach has been to model an input sentence using bi-directional Long Short-Term Memory Networks (bi-lstms) (Cross and Huang, 2016; Kiperwasser and Goldberg, 2016b). The result is a vector for each word that encodes both its information, and relevant information from other parts of the sentence. This approach enabled better results with fewer features than was possible before (Cross and Huang, 2016; Shi et al., 2017).

Another approach has been to represent the dependency tree itself with some form of recursive network, either bottom-up (Dyer et al., 2015; Kiperwasser and Goldberg, 2016a; Stenetorp, 2013), or top-down (Le and Zuidema, 2014).

In this paper we propose a new method of recursively modelling dependency trees using LSTMs, which we call Recursive Tree LSTMs. Our experiments show that this method of representation is very powerful, and can even be used as an additional layer of encoding over bi-lstm feature representation, which results in a more informative model. The final parser is capable of achieving competitive results with a feature set consisting of only the top two items on the stack, which is the smallest feature set for an Arc-Standard dependency parser used successfully to date.

2 Recursive LSTM Trees (RLTs)

We propose a method of representing a dependency tree as a single dense vector that results from the repeat application of an encoding mechanism to sequences of head-dependent pairs.

[*] Work done while at University of Birmingham.

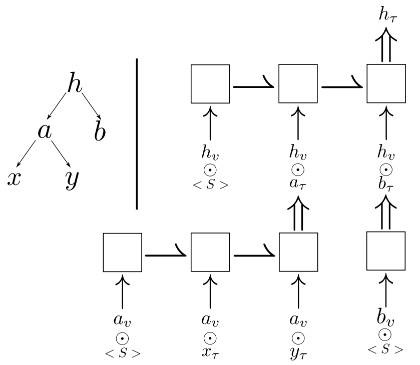


Figure 1: A compact representation showing how a subtree (left) is arranged as a sequence to produce a tree vector for the head token h_τ (right). The encoding mechanism here is a single forward LSTM. The operation $\odot$ is concatenation, $\upharpoonleft$ is input, $\rightharpoondown$ is the passing of the internal state from one LSTM step to another, and $\Uparrow$ is the output of the LSTM. $< S >$ represents the start tag, and is used both to signifiy the start of the head/child sequence and as a base case for leaf nodes, such as in the case of b_τ. The construction of x_τ and y_τ is done in the same way as for b_τ and is omitted for brevity.

Each node in the tree represents a head token's interaction with the representations of all of its immediate dependents. Similarly, the representations of each of these dependents are themselves the result of the interaction between their token and the representations of their corresponding dependents.

Each token has 2 representations, a vector representation v and a tree representation τ. The vector representation is the raw description of a token in its sentence, which in the most basic form can simply be the concatenation of the word and part-of-speech vectors of that token. However, contextualized representation has been shown to be a richer, more informative feature about a token and its position in a sentence (Cross and Huang, 2016; Kiperwasser and Goldberg, 2016b). We experiment with both approaches and confirm that a contextualized vector does improve the performance of RLTs, in addition to its properties being carried over to the RLTs themselves, meaning that parsing can be done with minimal features.

The tree representation of a token, on the other hand, encodes the dependency information of a token and its dependents. Consider a simple subtree consisting of a head token h and its dependents a and b as illustrated in Figure 1. The subtree is represented as a sequence of pairs of head vectors (h_v) and child trees (a_τ, b_τ). These pairs are then input to the encoding mechanism with the final output being the head tree vector h_τ.

The first pair in the sequence representation is always $(h_v, < S >)$, where $< S >$ has the same size as the output size of the encoding mechanism and represents the start tag of the sequence. This also serves as the base case for leaf nodes in the dependency tree as well as for tokens without dependents in partially built trees while parsing.

Each input pair uses the same h_v which is then concatenated with the tree representation of the dependent. The dependents are presented in their order of appearance in the sentence, and the encoding mechanism output at each step can be taken to represent the subtree of h including the dependents introduced up to that step. The recursive element of this formulation is the repeat application of the encoding mechanism, in a bottom-up approach, in order to produce tree representations for tokens that are then used in turn to produce the tree representations of their corresponding heads.

The head token h has one dependent who also has dependents and another which has none. b_τ is represented by the base case with $(b_v \odot < S >)$, while a_τ requires 2 additional steps to incorporate information from x_τ and y_τ.

The dependents a_τ and b_τ must be calculated first before h_τ can be produced, and by extension x_τ and y_τ are required first in order to calculate a_τ. In this way the final dependency tree representation is built recursively, bottom-up, with the final representation being the tree representation $ROOT_\tau$.

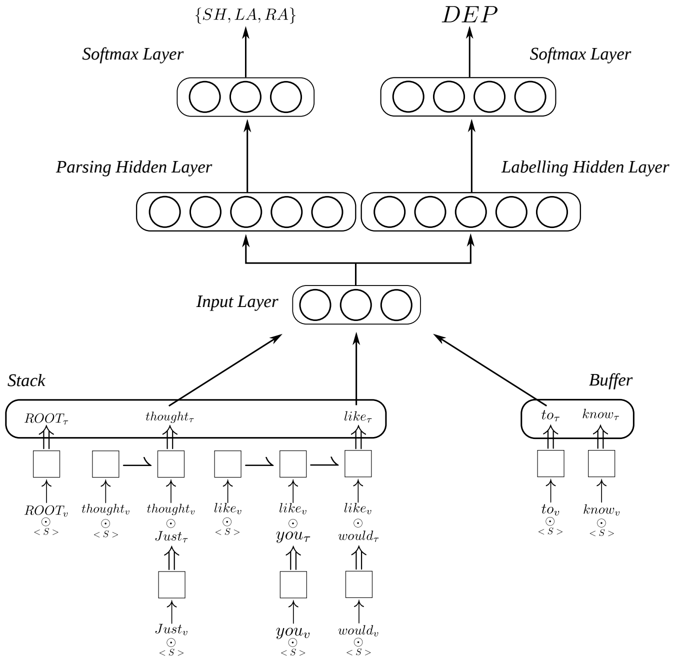


Figure 2: Example of a parser configuration with features using RLTs.

3 Implementation & Training Details

We implemented[1] our model in python using the DyNet framework (Neubig et al., 2017). The encoding mechanisms used by the RLTs in our experiments used 2 layers of LSTMs of size 256. For experiments using bi-lstm contextualized representation, we also used 2 layers of bi-lstms of size 256 in each direction. For the basic vector representations we used randomly initialized part-of-speech tag vectors of size 50, and for word embeddings we used vectors of size 100 initialized using the GloVe vectors (Pennington et al., 2014) trained on 6B. tokens with 400k vocabulary.

The tree vectors of relevant RLTs are then concatenated and passed as input to two sets of two feed-forward hidden layers of size 256, with rectified linear units (ReLUs) (Nair and Hinton, 2010) as activation functions. The two sets of hidden layers are responsible for modelling the relevant information for dependency parsing and dependency labelling separately, similar to the hierarchical architecture used by Cross and Huang (2016). We set a dropout rate of 0.3 on all LSTMs (Gal, 2015) and the hidden layers (Hinton et al., 2012). In our experiments we tried different dropout rates, but the differences were too small to experiment with separate dropout rates for different layers. The final output layers are two separate softmax layers with the same structure as in the setup of Cross and Huang (2016), in which the scores of the output layer corresponding to $\{SH, LA, RA\}$ uses the output of the dependency parsing hidden layers, and the output layer scoring dependencies $\{DEP\}$ uses the output of the dependency labelling hidden layers, where DEP is the set of all possible dependency labels. An illustration of the architecture of the parser is show in figure 2. All weights and pos tag vectors were initialised uniformly (Glorot and Bengio, 2010). For training we use a negative log likelihood loss function, $-\sum_i log(y_i)$, where y_i is the score of the gold transition from the final softmax layer for the training input/output pair i in the mini-batch. We use mini-batch updates of 10 sentences, and stop training after 30 epochs. We optimize the model parameters using Adam (Kingma and Ba, 2014) with a learning rate $\alpha = 1 \times 10^{-3}$. We train our models using the Wall Street Journal (WSJ) section from the Penn Treebank (Marcus et al., 1993). We use §2-21 for training, §22 for development, and §23 for testing. We use Stanford Dependencies (SD) (De Marneffe et al., 2006) converted from constituency trees using version 3.3.0 of the converter. As is standard we use predicted POS tags for the train, dev, and test sets. We report unlabeled attachment score (UAS) and labeled attachment score (LAS), with punctuation excluded. The models are tuned on

[1] Implementation available at `https://github.com/MohabElkaref/rlt`

Encoding Type	UAS	LAS
word/pos embeddings	92.94	90.61
contextualized vectors	**94.26**	**92.01**
K & G (2016a)	**93.3**	90.8
Dyer et al. (2015)	93.2	**90.9**

Table 1: Development set scores on WSJ (SD) comparing between h_v being a concatenation of the tokens word/pos vectors and h_v being a concatenation of contextualized vectors.

Feature Set	UAS	LAS
$\{s_{0-3}, b_{0-3}\}$	**94.26**	**92.01**
$\{s_{0,1}, b_0\}$	94.23	91.99
$\{s_{0,1}\}$	93.88	91.72

Table 2: Development set scores for different feature sets, using a bi-lstm contextualized vector as h_v, for Forward and Bi-directional encoding.

the development set, with the tuning that produced the highest UAS used to obtain the final scores on the test set. We additionally report results on the Universal Dependency set used in the CoNLL'18 shared task in Table 4 for Catalan, German, English, Spanish, French, Italian, and Norwegian.

4 Experiments & Results

For our initial set of experiments we trained models that used the top 4 RLTs on the stack, and the front 4 on the buffer as input features to the feed forward hidden layer. We compare our results initially to those of Dyer et al. (2015), who used Stack-LSTMs, and Kiperwasser and Goldberg (2016a), who used Hierarchical Tree-LSTMs, since they are the closest in the literature to our approach. We make a more complete comparison with state of the art Transition-based parsers in table 3.

Recursive representation was used by Dyer et al. (2015) to represent elements on the stack, similar to our approach. However, their representation is computed through the recursive application of a feed-forward composition function that encodes a (head, relation, dependent) tuple, encoding children in the order in which they are reduced. Kiperwasser and Goldberg (2016a) uses a bottom up recursive approach to build a tree representation as well, but separates the sequence of children into a left and a right sequence, with the head itself being the start of both sequences, and the final representation of the subtree being a concatenation of the output of both sequences. As in our work, Kiperwasser and Goldberg (2016a) use bi-LSTM vectors to represent words being input to the encoding LSTM.

When setting h_v to be the concatenation of the word and pos vectors, the resulting accuracy score, shown in table 1, approaches the performance of Dyer et al. (2015) and Kiperwasser and Goldberg (2016a). Using bi-lstm contextualized representation as h_v, however, significantly improves accuracy to 94.26/92.01 on the development set and beating both of our baselines.

Our second set of experiments were to investigate whether or not RLTs retain the properties of the bi-lstm representation in addition to its own, i.e., produce an h_τ that can represent a token's special position in a sentence *in addition to* representing it as the head of its own subtree.

The results shown thus far are the results of a wide feature set, the first 4 items on both structures $\{s_{0-3}, b_{0-3}\}$, which is comparable to earlier feature sets used by Weiss et al. (2015) and Chen and Manning (2014), but without the need for structural features, such as left-most and right-most dependents which are already encoded in the way a tree vector is produced. The results in Table 2 show the performance of our RLT models on increasingly small feature sets. This second set of experiments show that RLTs are also able to represent contextual information about the node from the bi-lstm layer in addition to its own structural information. Interestingly the drop in the accuracy of RLTs with the complete removal of buffer features is limited. Our minimal feature set here consists of only the top 2 items on the stack $\{s_{0,1}\}$. These 2 elements represent the fundamental task of an Arc-Standard parser, which is to decide whether or not these 2 words are related, and so are not themselves contextual features.

5 Discussion

Vector tree representation has a long history, primarily used to model constituency trees using Recursive neural networks (Goller and Kuchler, 1996; Socher et al., 2010). Such networks relied on the repeat application of a feed forward layer to encode a fixed maximum number of relations. Adapting this

	UAS	LAS
This work		
8 feats. + word/pos embeddings	92.72	90.55
8 feats. + contextualized vectors	94.13	92.11
2 feats. + contextualized vectors	94.04	91.93
Recursive Tree		
Le and Zuidema (2014)	93.84	91.51
Dyer et al. (2015)	93.1	90.9
Kiperwasser and Goldberg (2016a)	93.0	90.9
Ballesteros et al. (2016)	93.56	91.42
Feed Forward		
Chen and Manning (2014)	91.80	89.60
Weiss et al. (2015)	93.99	92.05
Andor et al. (2016)	**94.61**	**92.79**
Bi-lstm contextualized representation		
Cross and Huang (2016)	93.42	91.36
Kiperwasser and Goldberg (2016b)	93.9	91.9
Shi et al. (2017)	94.53	N/A

Table 3: Test set scores on WSJ (SD) for some of the highest scoring Transition-based Dependency Parsers in current literature. Contextualized vectors refer to the bi-lstm vector representation used for h_v, and word/pos embeddings refers to the concatenation of these vectors to represent h_v. 8 feats. refers to the use of the top 4 items on the stack and buffer, 2 feats. refers to the use of the top 2 items on the stack.

Corpus	UAS	LAS
`ca_ancora`	90.34	87.72
`de_gsd`	76.71	71.56
`en_ewt`	82.86	80.18
`es_ancora`	89.78	87.10
`fr_gsd`	84.15	80.04
`it_isdt`	90.45	88.22
`no_bokmal`	85.83	82.70

Table 4: Test set scores on 7 corpuses from the CONLL‘18 shared task. These sets use Universal Dependencies, and use an F1 score calculation for UAS and LAS that includes punctuation.

approach to an arbitrary number of dependents results in deep narrow trees and the gradient vanishing problem. One approach to deal with this has been the Tree-LSTM model, an amended gating mechanism proposed by Tai et al. (2015) based on LSTMs.

For transition-based parsing earlier work with recursive representation includes Stenetorp (2013), who uses a recursive layer to model dependency trees in a manner similar to that used in constituency parsers, but does not produce a high accuracy.

Our main comparisons have been with the work of Kiperwasser and Goldberg (2016a) as it is the closest to our work. They use a bottom up recursive approach to build a tree representation as well, but separate the sequence of children into a left and a right sequence, with the head itself being the start of both sequences, and the final representation of the subtree being a concatenation of the output of both sequences. As in our work, Kiperwasser and Goldberg (2016a) use bi-lstm vectors to represent words being input to the encoding LSTM. We note that in the case of dependents that are leaf nodes in the dependency tree, the representation of Kiperwasser and Goldberg (2016a) models the left sequence backwards and the right sequence forwards. They do not encode information considering the entire set of dependents.

We also compare our results with Dyer et al. (2015), who use a bottom encoding to represent words on the stack, and then uses a stack-LSTM to represent the stack and buffer. The main point of interest here is a recursive composition function which encodes a (head, relation, dependent) tuple, and represents heads with multiple dependents by reapplying the composition function with the previous output as the head. The dependents are encoded into this representation as they are added to the tree, which again means an unordered representation of dependents. Our models suffered a drop in accuracy when we used an unordered sequence of dependents, which could be a possible explanation for the $\sim 1\%$ difference in accuracy scores.

The performance of RLTs shows a considerable ability to encode structural information into a single dense vector. This ability is highlighted when comparing with Weiss et al. (2015), where the resulting accuracy scores are comparable but only with the additional representation of a structured perceptron. Similarly, the scores of Kiperwasser and Goldberg (2016b) improve by using structural features in addition to the initial set of $\{s_{0-2}, b_0\}$, with the left and right-most modifiers of the first 3 and the left-most modifier of the last, for a total of 11 contextualized features. In both of these cases the stack and buffer features are similar, with RLTs showing an ability to implicitly encode useful structural features in the final tree vector τ.

Additionally RLTs gain much from the use of contextualized vectors as the base representation v. The structure of RLTs predictably is not capable of modelling the sequential position of a word in its sentence, but it can retain the information modelled by the bi-lstm representation fed into it.

Finally, our model produces competitive results with a minimal feature set that, to the best of our knowledge, has not yet been achieved for Arc-Standard, but has been achieved for Arc-Eager and Arc-Hybrid by Shi et al. (2017). A key difference is that our minimal features set consisted of the top 2 items on the stack, while Shi et al. (2017) used the first items from the stack and buffer, which did not work for Arc-Standard. This difference could be due to the different definitions of the LA transition in particular which use the front of the buffer as head, while Arc-Standard limits all transition effects to the stack. Our results remain behind those of Andor et al. (2016) and Shi et al. (2017), both of whom used global loss function, in addition to the latter's exact decoding.

6 Conclusion & Future Work

In this work we proposed a recursive tree architecture capable of modelling both subtrees and whole dependency trees. This method exploits the ability of deep learning to model combinations of features as needed in dense vectors, moving further away from feature selection to more expressive architectures. The resulting vector representation for each word encodes information that describes its position in its dependency tree, as well as its sequential position in its original sentence.

Furthermore, the model might be useful for other applications as well, including question answering and sentence similarity, as the final dense representation captures entire sentences.

Acknowledgements

This work was part funded by the STFC Hartree Centres Innovation Return on Research programme, funded by the Department for Business, Energy Industrial Strategy.

References

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*.

Miguel Ballesteros, Yoav Goldberg, Chris Dyer, and Noah A. Smith. 2016. Training with exploration improves a greedy stack lstm parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2005–2010. Association for Computational Linguistics.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750.

James Cross and Liang Huang. 2016. Incremental parsing with minimal features using bi-directional LSTM. *CoRR*, abs/1606.06406.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*.

Yarin Gal. 2015. A theoretically grounded application of dropout in recurrent neural networks. *arXiv:1512.05287*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May. PMLR.

Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352. IEEE.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016a. Easy-first dependency parsing with hierarchical tree lstms. *arXiv preprint arXiv:1603.00375*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016b. Simple and accurate dependency parsing using bidirectional lstm feature representations. *arXiv preprint arXiv:1603.04351*.

Phong Le and Willem Zuidema. 2014. The inside-outside recursive neural network model for dependency parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 729–739, Doha, Qatar, October. Association for Computational Linguistics.

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

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Tianze Shi, Liang Huang, and Lillian Lee. 2017. Fast(er) exact decoding and global training for transition-based dependency parsing via a minimal feature set. *CoRR*, abs/1708.09403.

Richard Socher, Christopher D Manning, and Andrew Y Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–9.

Pontus Stenetorp. 2013. Transition-based dependency parsing using recursive neural networks. In *NIPS Workshop on Deep Learning*. Citeseer.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. *arXiv preprint arXiv:1506.06158*.

Hao Zhou, Yue Zhang, Shujian Huang, and Jiajun Chen. 2015. A neural probabilistic structured-prediction model for transition-based dependency parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1213–1222.

Improving the Annotations in the Turkish Universal Dependency Treebank

Utku Türk[‡], **Furkan Atmaca**[‡], **Şaziye Betül Özateş**[*],
Balkız Öztürk[‡], **Tunga Güngör**[*], **Arzucan Özgür**[*]
[‡]Department of Linguistics
[*]Department of Computer Engineering
Boğaziçi University
Bebek, 34342 İstanbul, Turkey
`utku.turk,furkan.atmaca,saziye.bilgin,`
`balkiz.ozturk,gungort,arzucan.ozgur@boun.edu.tr`

Abstract

This study focuses on a comprehensive analysis and manual re-annotation of the Turkish IMST-UD Treebank, which was automatically converted from the IMST Treebank (Sulubacak et al., 2016b). In accordance with the Universal Dependencies' guidelines and the necessities of Turkish grammar, the existing treebank was revised. The current study presents the revisions that were made alongside the motivations behind the major changes. Moreover, it reports the parsing results of a transition-based dependency parser and a graph-based dependency parser obtained over the previous and updated versions of the treebank. In light of these results, we have observed that the re-annotation of the Turkish IMST-UD treebank improves performance with regards to dependency parsing.

1 Introduction

With its unique set of tags and as a multilingual framework of natural language processing (NLP), the Universal Dependencies (UD) Project[1] offers researchers a common ground regarding the specific features of every language. However, not all languages contribute equally to this project. One language that has not been thoroughly studied is Turkish, which constitutes a canonical example of agglutinative language. Turkish bears a unique challenge in the pursuit of syntactic parsing due to its highly agglutinative morphology and flexible word order patterns. The inefficacy and the inaccuracies of the treebanks previously proposed for Turkish hinder the development of its use in NLP frameworks and its contribution to the UD Project. With this in mind, we set out to find a way to take Turkish Treebanks a step further to overcome the challenges Turkish poses.

In studies that have been previously conducted on Turkish treebanks (Sulubacak et al., 2016a; Sulubacak et al., 2016b; Pamay et al., 2015; Pamay and Eryiğit, 2014; Oflazer et al., 2003; Çöltekin, 2015), there is one very obvious drawback: They differ significantly from one another in terms of their compliance with the rules of Turkish grammar and the UD annotation patterns. While the ITU Validation set and the Turkish PUD treebank follow the finely grained rules of Turkish grammar and annotation guidelines of the UD Project, the IMST-UD Treebank has a coarsely grained structure and inconsistencies resulting from its automated creation. For example, the Turkish PUD treebank utilizes language specific syntactic relations, i.e., `obl:tmod`, `acl:relcl`, `det:predet`, and `flat:name`, quite generously, and the annotation of the Turkish PUD does not have problems that stem from nominalization-based morpho-ortographical similarities (Türk et al., 2019). Our objective is to unify the annotation patterns and decisions based on the requirements of Turkish grammar across the existing treebanks through manual annotations of all the sentences in the Turkish treebanks within the UD framework. By doing so, we aim to create a basis for a future treebank that we will build. The new treebank will consist of an additional ten thousand unique sentences. We have recently re-annotated the Turkish PUD Treebank with our proposed guidelines

[1] `https://www.universaldependencies.org`

(Türk et al., 2019). In this study, we unify the annotation scheme of the Turkish treebanks and manually re-annotate all the sentences of the IMST-UD Treebank.

2 IMST-UD Treebank

Following the initial efforts to build an annotated treebank, in English (Marcus et al., 1993) and in other languages, Atalay et al. (2003) and Oflazer et al. (2003) introduced a pioneering METU-Sabancı treebank for Turkish. Then, this treebank was re-annotated as IMST Treebank and automatically converted to the UD framework, which resulted in unrivaled scores in NLP tasks for Turkish (Sulubacak et al., 2016a). However, this re-annotation process did not include a linguistics team; instead, it consisted of only one linguist and a team of NLP specialists. Moreover, every annotator worked on their own share of sentences, which resulted in a highly disharmonious picture overall.

As an immediate result of the non-communicative nature of the annotation process, the IMST-UD treebank is incoherent on most of the items, with inconsistencies ranging from ongoing debates on the distinction between `obl` and `obl:arg`[2] in case-marking languages to the very simple concepts of `root` and `punctuation`. Nevertheless, the treebank's morphological segmentation and *inflectional groups* analysis, which refers to the morphosyntactic division of words with respect to their derivational morphology, made the IMST-UD Treebank one of the cornerstones of Turkish and Turkic treebank studies. For this study, we only included the IMST-UD Treebank in the re-annotation process. We have recently re-annotated the Turkish PUD Treebank (Türk et al., 2019) and plan to include the ITU Validation Set in the future with another brand new UD Treebank. We have excluded the Grammar-Book Treebank, created by Çöltekin (2015), as it may contain incomplete phrases and structures that may hinder the parser.

3 The Boğaziçi-ITU-METU-Sabancı Treebank (BIMST)

3.1 Overview

In this work, we have manually examined all the sentences in the IMST-UD treebank with the version UD 2.2 and updated the annotation of the syntactic relations and the head-dependency relations. We have used the most recent version of the treebank that is available on GitHub[3]. For this paper, we have excluded the improvement of the morphological segmentations in the treebank, and accepted the previous morphological parsing (Çöltekin, 2016). The reason for this exclusion is that we believe that the inconsistencies and the annotation problems demanded the most urgent attention. For this reason, we have accepted the same morphological segmentation principles that are used in the most recent version of the treebank.

3.2 Process

In the manual re-annotation process of the IMST-UD Treebank, we first reviewed the definitions of UD (Nivre et al., 2016) and the Stanford Dependencies work (De Marneffe et al., 2014) that influenced important components of the UD framework. Then, we compared our example sentences with the examples in the UD Project website for Turkish and also cross-linguistically. Having settled on the definitions, we reported errors and inconsistencies found in the existing treebanks. The errors and inconsistencies typically resulted from the automated nature of the IMST-UD treebank tags or were due to incorrect linguistic analyses.

To resolve the inconsistencies in the previous treebanks, a team of three linguists and three NLP specialists was formed for the current study. Moreover, discussions were held for potential solutions and their merits according to both Manning's Law (Nivre et al., 2017) and the necessities of Turkish grammar. The criteria we used took into consideration the fine grained distinctions established in Turkish linguistics, typological adequacy, ease of rapid and consistent annotation, ease of understandability, and high accuracy in parsing. These criteria helped us narrow down the list of solutions, and in the end, we decided to implement the most feasible ones.

[2] This change follows from the recent discussion that can be found in https://universaldependencies.org/workgroups/core.html.

[3] https://universaldependencies.org/treebanks/tr_imst/

All the revisions made in the IMST-UD Treebank were recorded and, then were incorporated into the CONLL-U files using the udpipe package in R (Wijffels et al., 2018). The annotated treebanks, the detailed history of changes made in the annotation process, and our proposed new guidelines are available at `https://github.com/boun-tabi/UD_Turkish-BIMST`.

3.3 Revision

In the IMST-UD Treebank, we re-annotated a total of 5,635 sentences. Table 1 depicts the items that we altered most within the IMST-UD Treebank. In addition to these changes, we introduced 8 previously unused dependency types.

Previous Treebank	**Updated Treebank**	**Number of Alterations**
nmod	advcl	666
obj	ccomp	481
nmod:poss	nsubj	312
obj	nsubj	276
obl	iobj	194
obl	obl:arg	137

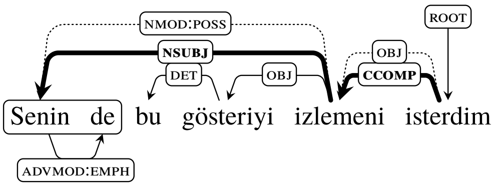
Table 1: The number of alterations that we made for the most frequent changes.

3.3.1 Transparency of Embedded Clauses

The most significant group of changes made in the annotation was based on the inadequate analysis of the internal structure of nominalized embedded clauses in the IMST-UD Treebank. In terms of their external syntax, such clauses behave like regular nouns; in fact, linguistic tests, such as replacement, would deem the whole embedded structure a noun, rather than a clausal object (Göksel and Kerslake, 2005).

However, these structures maintain their predicate's argument structure and inner hierarchy between their own dependents, having the typical properties of a clause in terms of their internal syntax. With this in mind, the outlined genitive phrase was erroneously marked as a possessor in the IMST-UD Treebank as indicated by the dotted lines in sentence (1), even though it does not establish a possessive relation, but syntactically is the genitive marked subject of the nominalized embedded clause (Göksel and Kerslake, 2005). Thus, as shown by the bold line[4] in sentence (1)[5] we marked it as `nsubj`.

(1)



Sen-in de bu gösteri-yi izle-me-ni iste-r-di-m.
you-gen too this show-acc watch-nmlz-poss want-aor-pst-1sg

"I would have wanted you to watch this show, as well."

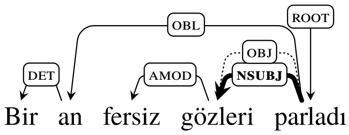
One other issue regarding sentence (1) is that morpho-phonologically identical outputs, the genitive marker on the possessor and the genitive on the subject of embedded sentences, are rendered as possessing the same syntactic function. One other morpho-phonological ambiguity stems from the possessor morpheme in Turkish, which can be ambiguous with the accusative case when it follows a word-final consonant. As seen in sentence (2), *gözleri* is the subject of the unaccusative verb *parla-*. However, it is marked with the suffix *-i* which can be either the possessive suffix on the possessee on nominals or the accusative case. In the IMST-UD Treebank, sentences such as sentence (2) are erroneously marked as `obj` due to the ambiguity mentioned above. However, the ambiguity between the accusative case and the

[4] In all dependency trees in this paper, the dotted lines show the syntactic relations used in the IMST-UD Treebank, the bold ones indicate the re-annotated ones in the updated treebank, and the fine lines show unaltered dependencies.

[5] Abbreviations used in this paper are as follows: 1 = first person, abl = ablative, acc = accusative, aor = aorist, caus = causative, cvb = converb, dat = dative, gen = genitive, neg = negative, nmlz = nominalizer, pl = plural, poss = possessive, pst = past, sg = singular.

possessive suffix is resolved by replacing *gözler* with a noun that has a word-final vowel as in sentence (3). In such environments, the accusative case surfaces as *-yi* as in sentence (1), and the possessive morpheme surfaces as *-si*.

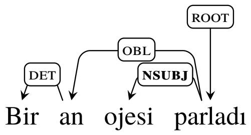
(2)



Bir an	*fersiz*	*göz-ler-i*	*parla-dı.*
one moment	dull	eye-PL-POSS	sparkle-PST

“Her dull eyes sparkled for a moment."

(3)

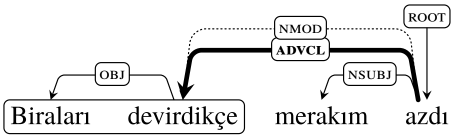


Bir an	*oje-si*	*parla-dı.*
one moment	nail.polish-POSS	sparkle-PST

“Her nail polish sparkled for a moment."

Another example of the transparency issue within the embedded clauses is the case of converbs and adverbials. In Turkish, these structures also retain their sentential characteristics, which allow them to behave like an embedded clause. This process is not unique to Turkish, and in fact, the English gerund structure is another example of this phenomenon. This is why we, again, ignored the morpho-phonological similarities. Instead of `nmod`, we annotated such items as `advcl`, and the same applied for subjects of embedded structures, which were annotated as `nmod:poss` and we re-annotated them as `nsubj` since they are a core argument for the embedded structure. The new analysis of embedded clauses and core dependents in this section also follows what is proposed by Przepiórkowski and Patejuk (2018) regarding the openness and transparency of an argument structure.

(4)



Bira-lar-ı	*devir-dik-çe*	*merak-ım*	*az-dı.*
beer-PL-ACC	topple-NMLZ-CVB	curiosity-POSS.1SG	get.wild-PST

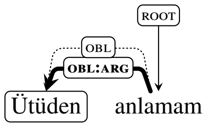
“As I finish my beers, my curiosity peaked."

As indicated by the dotted lines, sentence (4) is also misannotated as `nmod` in the IMST-UD Treebank, even though it is a nominalized converbial structure whose internal argument structure is explicit. We have marked such clauses as `advcl` in accordance with their syntactic function as indicated by the bold line in (4).

3.3.2 Core vs. Non-core Dependents

Another significant group of changes (*n=139,* n_{total}*=7,899*) made in the re-annotation was based on the definition of core arguments. In addition to canonical object case i.e., accusative case, Turkish also makes use of non-canonical case marking, such as dative, ablative, and locative, for marking obligatory object arguments. The same set of non-canonical case marking can also be used for adjuncts in Turkish. When those cases are selected lexically by the verb, the relation between the nominal head and the verbal head remains the same as if the case on the nominal head was the accusative case. However, non-canonically marked core arguments and adjuncts are both tagged as `obl` in the IMST-UD Treebank. This indicates that non-canonically case-marked arguments, which are obligatory to the sentence, are non-core dependents (De Marneffe et al., 2014). In our analysis, we differentiated the non-core adjuncts from the arguments by following an annotation scheme proposed by Zeman (2017). Zeman (2017) proposes a new syntactic relation for non-canonically marked core arguments, namely `obl:arg`, and differentiates between core objects and oblique arguments, which are also core elements in the sentence but marked with non-accusative case.

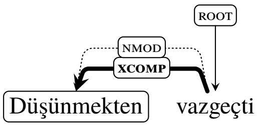
(5)



Ütü-den
ironing-ABL
anla-ma-m.
understand-NEG-1SG

“I do not know a thing about ironing."

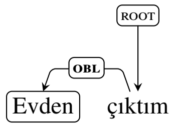
(6)



Düşün-mek-ten vazgeç-ti.
thinking-NMLZ-ABL give.up-PST

“She decided not to think about it."

(7)



Ev-den
home-ABL
çık-tı-m.
leave-PST-1SG

“I have left the home."

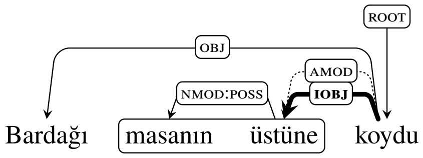
In sentences (5) and (6) above, a lexically selected non-canonical case, i.e. ablative, is used to mark core argument dependency. Ablative case, when not lexically selected, is attached to a non-core adjunct and indicates a semantically transparent meaning, such as source, departure, or cause (Erguvanlı-Taylan, 2015) as in sentence (7). However, in sentences like (5) it does not contribute to the meaning of the verb *anlamak* (meaning ‘to understand’) in any way, showing that it is lexically selected.

In sentence (6) on the other hand, the outlined constituent is not a nominal, but a nominalized non-finite clause, which is again a core argument. Nominalized clauses are formed with a subordinating suffix, as in item (6), but again are marked with the non-canonical ablative case (Göksel and Kerslake, 2005).

3.3.3 Introduced Dependency Types

In order to increase the linguistic adequacy and the efficiency of NLP tasks, we also introduced dependency types that were previously unused in the IMST-UD Treebank. We propose `advcl` and `iobj` as in sentences (4), (8), and (9), emulating other Turkic treebanks (Tyers et al., 2017). Moreover, we have introduced six more universal syntactic dependencies, namely `xcomp`, `dislocated`, `orphan`, `clf`, `goeswith`, and `dep`.

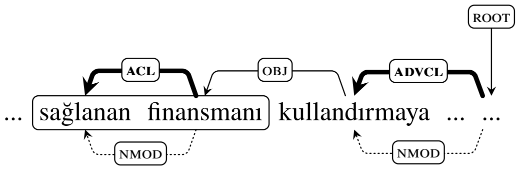
(8)



Bardağ-ı masa-nın üst-ün-e koy-du.
glass-ACC table-GEN top-POSS-DAT put-PST

“She put the cup on the table."

(9)



... *sağla-nan finansman-ı*
... supply-NMLZ financing-ACC
kullan-dır-ma-ya ...
use-CAUS-NMLZ-DAT ...

“... to make use of the funding provided ..."

In sentence (8), the previous treebank in the UD Project inaccurately marks the relation between the verb and *masanın üstüne* as `amod` since the word group does not modify any noun and, thus, is not an adjectival modifier. One may argue that it is a destination or goal and can be marked as `obl` syntactic relation. However, this analysis would again mean that the word group is not obligatorily required by the event structure of the verb. Hence, we decided to add the indirect object relation to the Turkish Treebank.

4 Experiments

In order to observe the effect of the changes on the parsing performance of the IMST-UD Treebank, a transition-based LSTM dependency parser (Özateş et al., 2018), which is a morphologically enhanced version of Ballesteros et al. (2015) and a state-of-the-art graph-based neural parser (Dozat et al., 2017) are trained on the previous and updated versions of the treebank separately. Both projective and nonprojective dependencies are included in the training and test phases, in contrast to many past studies on the IMST-UD Treebank as well as on its previous versions that used only the projective dependencies (Eryiğit and Oflazer, 2006; Eryiğit et al., 2008; Sulubacak et al., 2016b; Sulubacak et al., 2016a; Sulubacak and Eryiğit, 2018).

[5] 1,022 `advcl` , 351 `iobj` , and a total of 79 for `xcomp`, `dislocated`, `orphan`, `clf`, `goeswith`, and `dep`.

The training part of both versions of the treebank includes 3,685 annotated sentences and the development and test parts include 975 annotated sentences each. That is, we used the original training/development/test partition of the treebank in all of our experiments.

As the pre-trained word vectors, we used the Turkish word embeddings of the CoNLL-17 pre-trained word embeddings from Ginter et al. (2017).

In the evaluation of the dependency parser, we used word-based unlabeled attachment score (UAS) and labeled attachment score (LAS) metrics, where the UAS is measured as the percentage of words that are attached to the correct head, and the LAS is defined as the percentage of words that are attached to the correct head with the correct dependency type.

4.1 Results and Discussion

Table 2 shows the UAS and LAS F1-scores for words achieved by the parsers on the previous version and the updated version of the IMST-UD Treebank, namely the BIMST Treebank.

		IMST-UD	BIMST
Transition-based parser	**UAS**	65.91	**68.66**
	LAS	59.06	58.98
Graph-based parser	**UAS**	71.55	**75.49**
	LAS	64.86	**65.53**

Table 2: UAS and LAS scores of the two parsers on the previous and updated versions of the IMST-UD treebank.

From the experiment results, we observe that the performances of the parsers in finding correct head-dependent relations increase on the updated version of the IMST-UD Treebank. Although the number of unique dependency tags increased from 33 to 41 with the newly introduced 8 dependency types mentioned in the previous section, the labeled attachment score remains almost the same on the updated version for the transition-based parser and increases for the graph-based parser. Sentence (9) shows an example sub-sentence from the previous version of the treebank and its correct annotation in the updated version. Previously, the dependency tag between *sağlanan* and *finansmanı* was `nmod` although the appropriate tag would be `acl`. The trained parsers predict this dependency tag as `acl`. However, this prediction is counted as false when the previous treebank version is used. In the updated version, such errors were corrected, leading to a better accuracy in terms of parsing of the treebank.

5 Conclusion and Future Work

In this paper, we illustrated the issues in the previous Turkish treebanks, namely the inconsistencies in the annotation and the mismatches between the UD guidelines and the sentences from the treebanks. We also explained our most prominent changes in the re-annotation of the IMST-UD Treebank. We increased the number of syntactic dependency relations that are used to 41, following the UD guidelines more rigidly.

The results demonstrate that the changes made improved the parsing performance with respect to the UAS metric for both of the parsers and the LAS metric for the graph-based parser. Although previously unused dependency relations are included, which pose a challenge for the parser in labeling the dependency relations, there was only a minor decrease in LAS score for the transition-based parser.

As future work, we aim to build the next Turkish treebank on a solid basis. We believe that a new and much more extensive treebank based on a more nuanced and up-to-date corpus should be our trajectory so that Turkish treebanking efforts within the UD framework and NLP processes will yield more accurate results.

Acknowledgments

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant number 117E971 and as a graduate scholarship.

References

Nart Bedin Atalay, Kemal Oflazer, and Bilge Say. 2003. The Annotation Process in the Turkish Treebank. In *Proceedings of 4th International Workshop on Linguistically Interpreted Corpora (LINC-03) at EACL 2003*.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. 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*, pages 349–359. Association for Computational Linguistics.

Çağrı Çöltekin. 2015. A Grammar-Book Treebank of Turkish. In Markus Dickinson, Erhard Hinrichs, Agnieszka Patejuk, and Adam Przepiórkowski, editors, *Proceedings of the 14th Workshop on Treebanks and Linguistic Theories (TLT 14)*, pages 35–49.

Çağrı Çöltekin. 2016. (When) do We Need Inflectional Groups? In *Proceedings of The First International Conference on Turkic Computational Linguistics*.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford Dependencies: A Cross-linguistic Typology. *In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 4585–4592.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30.

Eser Erguvanlı-Taylan. 2015. *The Phonology and Morphology of Turkish*. Boğaziçi Üniversitesi.

Gülşen Eryiğit and Kemal Oflazer. 2006. Statistical Dependency Parsing for Turkish. In *11^{th} Conference of the European Chapter of the Association for Computational Linguistics*.

Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. 2008. Dependency Parsing of Turkish. *Computational Linguistics*, 34(3):357–389.

Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 Shared Task - Automatically Annotated Raw Texts and Word Embeddings. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Aslı Göksel and Celia Kerslake. 2005. *Turkish: A Comprehensive Grammar*. Comprehensive grammars. Routledge.

Mitchell Marcus, Beatrice Santorini, and Mary Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: the Penn Treebank. 19:330–331.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis M. Tyers. 2017. Tutorial on Universal Dependencies. Presented at European Chapter of the Association for Computational Linguistics, Valencia [Accessed: 2019 04 08].

Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. 2003. Building a Turkish Treebank. In *Treebanks, Building and Using Parsed Corpora*, pages 261–277.

Şaziye Betül Özateş, Arzucan Özgür, Tunga Güngör, and Balkız Öztürk. 2018. A Morphology-based Representation Model for LSTM-based Dependency Parsing of Agglutinative Languages. *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 238–247.

Tuğba Pamay and Gülşen Eryiğit. 2014. ITU Validation Set for Metu-Sabancı Turkish Treebank. In *Proceedings of the TURKLANG'14 International Conference on Turkic Language Processing*, Istanbul, 06-07 November.

Tuğba Pamay, Umut Sulubacak, Dilara Torunoğlu-Selamet, and Gülşen Eryiğit. 2015. The Annotation Process of the ITU Web Treebank. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 95–101.

Adam Przepiórkowski and Agnieszka Patejuk. 2018. Arguments and Adjuncts in Universal Dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3837–3852.

Umut Sulubacak and Gülşen Eryiğit. 2018. Implementing Universal Dependency, Morphology, and Multiword Expression Annotation Standards for Turkish Language Processing. *Turkish Journal of Electrical Engineering & Computer Sciences*, 26(3):1662–1672.

Umut Sulubacak, Memduh Gökırmak, and Francis M. Tyers. 2016a. Universal Dependencies for Turkish. *Proceedings of the 26th International Conference on Computational Linguistics (COLING-16)*, pages 3444–3454.

Umut Sulubacak, Tuğba Pamay, and Gülşen Eryiğit. 2016b. IMST: A Revisited Turkish Dependency Treebank. In *Proceedings of 1st International Conference on Turkic Computational Linguistics, TurCLing*, pages 1–6.

Francis M. Tyers, Jonathan Washington, Çağrı Çöltekin, and Aibek Makazhanov. 2017. An Assessment of Universal Dependency Annotation Guidelines for Turkic Languages. Tatarstan Academy of Sciences, 10.

Utku Türk, Furkan Atmaca, Şaziye Betül Özateş, Abdüllatif Köksal, Balkız Öztürk, Tunga Güngör, and Arzucan Özgür. 2019. Turkish Treebanking: Unifying and Constructing Efforts. In *Proceedings of the 13th Linguistic Annotation Workshop (LAW XIII)*, pages 166–177, Florence, Italy, August 1, 2019.

Jan Wijffels, Milan Straka, and Jana Strakov, 2018. *udpipe: Tokenization, Parts of Speech Tagging, Lemmatization and Dependency Parsing with the 'UDPipe' 'NLP' Toolkit*. BNOSAC, Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic.

Daniel Zeman. 2017. Core Arguments in Universal Dependencies. *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 287–296.

Towards Transferring Bulgarian Sentences with Elliptical Elements to Universal Dependencies: Issues and Strategies

Petya Osenova
LMaKP
IICT-BAS
Sofia, Bulgaria
petya@bultreebank.org

Kiril Simov
LMaKP
IICT-BAS
Sofia, Bulgaria
kivs@bultreebank.org

Abstract

The paper considers the problems in transferring the sentences with elliptical elements from the original BulTreeBank into the Universal Dependencies style. The similarities and differences between the original constituency annotation scheme and the target dependency one are outlined to show that the current UD scheme needs elaboration to capture more complex cases.

1 Introduction

BulTreeBank (BTB) — an HPSG-based treebank of Bulgarian (Simov et al., 2005) — encodes both constituent and head-dependant structure in each phrase.[1] This facilitates the conversion from a constituent to a dependency representation. It should be noted, however, that BTB is not a typical HPSG treebank per se. It reflects the main principles of this theory and makes use of the structure sharing mechanism (especially for encoding phenomena like control) but it does not represent complex feature structures. The detailed features and their interaction are encoded within the node labels like VPS for verbal head-subject phrase and the structure of the constituent tree.

The current conversion of the treebank into the Universal Dependencies (UD) annotation scheme does not include the sentences with elliptical elements. These sentences amount to 1007 altogether. In the recent release editions most of the enhanced dependencies have already been added. The enhanced dependencies include the following phenomena: null nodes for elided predicates; propagation of conjuncts; additional subject relations for control and raising constructions; arguments of passives (and other valency-changing constructions); coreference in relative clause constructions and modifier labels that contain the preposition or other case-marking information.

However, the sentences with ellipsis (including the null nodes for elided predicates in the list of enhanced dependencies above) are still not present in the resource. These sentences constitute about 7% of the treebank. Needless to say, they are very important because they illustrate frequent structures that are typical for the organization of Bulgarian sentences.

In this paper the annotation typology of ellipsis in BulTreeBank is presented together with discussion on strategies for transferring these annotations into the UD framework. We also discuss the complexity of the envisaged transfer with respect to the specific types.

Our submission is expected to contribute to the discussion on the proper handling of elliptical phenomena, especially when transferring them from a constituency to a dependency-based treebank. Despite the fact that it focuses on Bulgarian language only, we believe that our considerations would be useful also for modeling ellipsis in treebanks of other languages.

The structure of the paper is as follows: in the next section related work is outlined briefly. Section 3 focuses on modeling ellipsis in the original BulTreeBank. Section 4 compares the annotation of elliptical phenomena in the original treebank with the strategies within UD. Section 5 concludes the paper.

[1] Coordination is an exception. It is considered as non-headed phrase.

2 Related Work

There is extensive literature on the ellipsis and its treatment in one or more languages, on grammatical and cognitive levels, etc. Here, however, only few findings will be mentioned that mainly focus on annotations within dependency-based frameworks.

(Mikulova, 2014) presents the typology of ellipsis in Czech in the dependency theory of Functional Generative Description. Since this is a multistratal theory and thus distinguishes between surface and deep levels, ellipsis is mainly modeled on deep (tectogrammatical) level. Within the scope of the surface ellipses the author includes the so-called'structural ellipses' (type 1). Here belong the following subtypes: a) ellipsis of the governing verb (I like coffee, but you [like] tea) and b) ellipsis of governing noun (Central [Europe] and Eastern Europe). Within the scope of the deep syntactic ellipses the author includes the so-called 'valency ellipses' (type 2). These include phenomena like textual ellipsis, general argument, control and reciprocity. While the former type (1) is analyzed with the insertion of an empty node, most cases that belong to the latter type (2) are marked with coreference arrows. In BTB the treatment of ellipsis is only on one level. Also, 'structural ellipses' group includes functional words/dependants (auxiliaries, modals, prepositions, etc.). 'Valency ellipses' are treated either with an insertion of a node (textual ellipsis), or with coreference (control), or not approached at all (general argument, reciprocity) i.e. a mixed strategy has been followed.

Another approach on the same language – Czech – is adopted by (Jelinek et al., 2015). The authors propose a constituent-based analysis for handling ellipsis, because it includes more information than the dependency-based one and also restores the syntactic structures. The constituent structures are output from the conversion of the dependency tree parses.

In a more theoretical paradigm is the survey of (Osborne and Liang, 2015). The authors used the dependency-based notion of catena to prove that in spite of the differing types of ellipsis in English and Chinese, the mechanism of analysis is equally suitable. Thus, the preferences of a language to certain types are made visible.

(Schuster et al., 2017) give arguments in favor of introducing distinct nodes for gapping constructions in the enhanced representation of UD guidelines version 2, instead of the previously used relations *remnant* and *orphan*. The similarity with the BTB approach is that they applied a recovery procedure to verbal ellipses. The difference is that in their case one node substituted the head verb and all its missing dependants, whereas in our case various recovery nodes are provided for the head and the dependants.

(Droganova and Zeman, 2017) discuss the varieties in the annotation of ellipsis within the UD treebanks. Their focus is on the dependent promotion when a head is elided. In the statistics survey of ellipsis in 41 treebanks (Table 1, p. 51) Bulgarian is given with a nearly zero representation of orphans (3/2) which is true given the fact that no sentences with ellipsis have been added in the releases. An example sentence is commented on Fig. 10 where the appearance of orphan is considered an error. In this particular case of introducing *orphan* instead of conjoining the two definite adjectives to the noun head, an error occurs because of the colloquial nuance (see the gloss: peaceful-the, political-the efforts). However, this nominal structure typically introduces a new entity. For example, in 'Bulgarian-the and Greek-the government' the governments are actually two referents.

There is also a line of work that describes the steps and challenges when transferring the linguistic information from the Lexical Functional Grammar (LFG) to UD — (Przepiorkowski and Patejuk, 2019). The authors chose to use the f structure for forming dependencies. Their transformation processes include the following stages: moving from LFG to LFG-like dependencies and then – rearranging dependencies. All these processes are not trivial. As it was mentioned before, our treebank is HPSG-based and HPSG-inspired. Thus our conversion started from the constituent structure, enriched with dependencies. Even from such a starting point our transformation suffered from lost information similarly to the one reported in (Przepiorkowski and Patejuk, 2019). This is for example, pro-dropness, control relations, etc.

3 Modeling of Ellipsis in the original BulTreeBank

In the original treebank that is constituency-based, the ellipsis is viewed as an expression that lacks an overt element. This element, however, is presupposed and thus recoverable or easily predicted by the context. Context might refer to many other phenomena: mostly grammar-based ellipsis (as pro-drop), optional arguments (as missing complements of transitive verbs), coreference and anaphora (especially in long texts and wider — including extralinguistic — discourse). Context can also be viewed as local and global.

The complexity of modeling ellipsis is also due to its close relatedness to and interference with linguistic phenomena like coordination and substantivization. The first one often licenses the insertion of recovery nodes in the position of the missing element, while the second licences the strategy of promoting dependants into heads when some head is missing.

As it was mentioned above, in BTB recovery markers for ellipsis were consistently added explicitly for all the modeled elliptical phrases. Thus, the idea is to preserve full syntactic structures.

Ellipsis was introduced through a mechanism of adding a special artificial node at the place of ellipsis, and connecting it with an index to the overt corresponding part (if there is such a part) or connecting it at the sentence level only (if the ellipsis is recoverable in a broader context or from world knowledge). Ellipsis was indicated on two levels: a) syntactic (V-Elip, N-Elip, A-Elip, PP-Elip, Prep-Elip) and b) discourse (VD-Elip, ND-Elip, PrepD-Elip). Please note that verbal ellipsis was briefly discussed in (Osenova and Simov, 2018) in relation to handling enhanced dependencies.

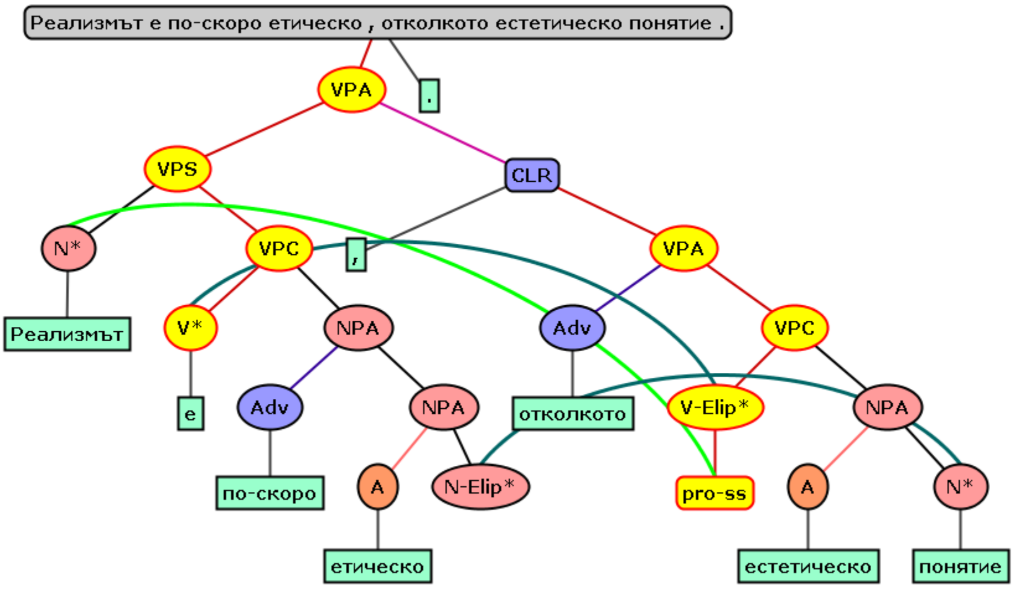


Figure 1: An example of ellipsis as encoded in the original HPSG treebank.

Fig. 1 depicts the representation of the sentence Реализмът е по-скоро етическо, отколкото естетическо понятие. “Realizmyt e po-skoro etichesko, otkolkoto estetichesko ponyatie.” (Realism is rather an ethical than an aesthetic concept.). The sentence contains the two main types of ellipses in BTB — verbal and nominal ones. In both cases an empty node is inserted where the elided element has to be present. Thus, in the example there is an ellipsis of the copula and an ellipsis of the noun “concept”. In addition to the verbal ellipsis node there is an additional node `pro-ss` representing the unexpressed subject which is coreferent with the subject within the first clause.

In Table 1 it can be seen that the most frequent type in BulTreeBank is the syntactic nominal ellipsis (N-Elip), immediately followed by the syntactic verbal ellipsis (V-Elip). It is interesting to note that the frequencies of the syntactic and the discourse verbal types are quite similar. The prepositional and adjectival types are rare.

However, it should be kept in mind that since ellipsis is related to other phenomena like substantivization and coordination, in some cases an alternative strategy might have been preferred. These interfering cases are discussed in more detail below.

In the annotation guidelines the coordination has been modeled at the following two levels: lexical and clausal. In both cases the following requirement is posited: if some heads are coordinated, they have to select the same arguments; similarly, if some dependants are coordinated, they have to be selected by the same head.

Substantivization being a promoter of dependants to heads, is used in very limited cases, such as: the missing head refers to general referents (the sick [people]; the three [men], etc.) or in systematic cases (one [student] of the students).

Type of Ellipsis	**Occurrences**
N-Elip	327
V-Elip	262
VD-Elip	255
ND-Elip	70
PP-Elip	12
Prep-Elip	3
PrepD-Elip	2
A-Elip	1

Table 1: Ellipsis Types in the original BTB and the number of their occurrences.

First, let us give some examples for each of the four most frequent types and briefly discuss them – N-Elip, V-Elip, VD-Elip and ND-Elip (examples 1–4). Then more complex cases are considered as well (examples 5–8).

In example (1) the nominal ellipsis is related to nominal phrasal coordination since the definite article in both adjectives (as indicated in the gloss) shows that the entities are different. Thus adjectival coordination is blocked here.

(1) Също така е развита химическата и техническата индустрия .
Also such is developed [chemical-the **N-Elip**] and [technical-the **industry**] .

‘Also, the chemical and technical industries have been developed.’

In example (2) verbal ellipsis occurs in the second conjunct of a clausal coordination. The negative particle had not been promoted into a verbal head. Instead, a null node was inserted.

(2) Иван отиде в градината , но Петър не .
Ivan **[went in garden-the]** , but [Peter not **V-Elip**] .

‘Ivan entered the garden but Peter did not.’

Example (3) shows a case where the verb ‘to be’ in present tense is missing. This characteristics is typical for the titles in newspapers. This ellipsis is considered as a kind of discourse ellipsis of subtype ‘exist’ instead of being analyzed as a truncated construction. The annotation repeats the one for the non-auxiliary verbs, namely inserting an artificial node as a place holder of the missing element.

(3) Социалните центрове пред стачка .
Social centers **VD-Elip [are]** before strike .

‘Social centers about to strike.’

In example (4) the discourse nominal ellipsis can be recovered only on the level of the whole article as well as from the local social knowledge. The elided element is the word for Bulgarian currency ‘levs’. Again, instead of promoting the numeral as a head, the nominal phrase structure is preserved through the addition of the node ND-Elip.

(4) Дори и с 5000 не бих се чувствала богата .
Even and with 5000 **ND-Elip [levs]** not would se.REFL felt rich .
‘Even with 5000 I would not feel rich enough.’

Let us now consider some more complex cases. In example (5) two things are interesting in the clausal coordination structure както..., така и...(as..., in a such a way...), ‘Similarly to X, Y did something’. First, the nominal coordination of the nominal subjects — Vulgaris and prince Mihaylo — is not eligible, because in this case the verb (crave for) agrees only with prince Mihaylo. Even if the verb was in plural agreement, adding more dependants around the heads – Vulgaris and prince Mihaylo – would prevent the coordinating of subjects only. Thus, a clausal coordination with restored ellipsis of the whole verbal phrase ‘was craving for Bulgarian land’ in the first clause is needed.

(5) Както Вулгарис , така и княз Михайло
As Vulgaris **VD-Elip [craved for Bulgarian lands]** , such and prince Mihaylo
ламтеше за български земи .
craved for Bulgarian lands .
‘Similarly to Vulgaris, prince Mihaylo was craving for Bulgarian land.’

In example (6) the problem is that the elided verb (there was) is a lexical opposite to the overt one (there was not). Thus, it is not enough just to copy the structural node, but is necessary to also indicate its opposite meaning. In BulTreeBank this was managed by providing subtypes of ellipses: identity in morphology and meaning; difference only in the morphological form; and change of the meaning into its opposite.

(6) Там нямаше заплаха, а само радост .
There **was**-not threat, but **V-Elip [was]** only joy .
‘There was not any threat, but only joy.’

In example (7) a valency-based ellipsis is introduced. The form *да отиде (to go-he)* is missing in the frame ‘I order someone to do something’. The overt past form of ‘went’ is not morphologically identical to the elided structure ‘to go’.

(7) Наскоро замина където му бяха наредили .
Recently **went-he** where him were ordered **V-Elip [to go]** .
‘Recently he went to where he was ordered to go.’

One case that might be reconsidered in favor of UD strategy, is presented in example (8). Here the structure of the NP has been preserved by inserting an N-Elip node. However, such cases might be extended into the application of substantivization strategy:

(8) Който се грижи за хорските работи, хората се грижат за неговите
Who se.REFL cares about people’s **things**, people se.REFL care about his
.
N-Elip [things] .
‘Who takes care of people’s business, people take care about his [business].’

From all the above illustrations it becomes clear that in the original BTB the goal was to maximally restore the clausal structure. Concerning the competition with coordination, the cases were solved with predefined structures that can coordinate only if they have the same selectional restrictions (from both points of view - being heads or being dependants). Concerning substantivization, it might be extended beyond the initially defined cases. There are also cases where the elided material is not related only to phenomena like coordination or substantivization, but also to phenomena like complementation (example 7) or free relatives (example 8).

4 Considering the sentences with ellipsis in UD framework

UD proposes the following strategies for handling ellipsis: a) a surface-based one (in which a special *orphan* relation is used) and b) a recovery-based one (in which null elements for the elided material are used – as in the enhanced dependencies) or promotion from the elided head to its dependants (when present) is introduced. The former relation adheres to surface syntax and thus – to truncated phrases where, in the absence of the head, non-typical relations connect their dependents (ex. in the sentence *John drinks water and Maria wine*, *Maria* and *wine* would be connected by the orphan relation). The latter relations are closer to the strategy that was adopted in the original BTB.

As it became clear already, in BTB the ellipsis has been always recovered, i.e. in this respect it followed somewhat a non-surface-like analysis. A full syntactic analysis of the structures was aimed, thus not considering the idea of having truncated phrases unless they form a typical constituent phrase (eg. [NP Good job!]).

The first type of the UD enhanced dependencies, called 'Null nodes for elided predicates', involves the addition of special null nodes in clauses with an elided predicate. An illustration of this idea is the following sentence: *I go to Varna, and you [V-Elip - go] to Sofia.* In Bulgarian V-Elip differs from the overt element by the category of person only. With this ellipsis recovery, the grammatical relations are maintained also in clauses without an explicit predicate. In BTB such predicates are introduced as V-Elip nodes in an appropriate place in the structure. Thus, this label can be mapped directly into the so-called null nodes. There are two cases of usage of V-Elip — representation of elided single verbal form; and representation of elided phrase — VP-ellipsis. The first case is the more straightforward one. In the second case in UD we need to introduce several null nodes in order to represent the whole VP. In addition to the null nodes in BTB also some variation of the grammatical features are encoded such as change in the number, tense, etc. For the moment it is not clear how to represent this in UD — see example 6. In such cases we encode the modified grammatical features as such for the null nodes.

Also, in UD each verb in a verbal complex is marked with a null node, while in BTB there is only one such substitute node for all the elided material. The principle is that the introduced recovery node refers to the maximal material that is elided.

In contrast to V-Elip, the null nodes annotated with VD-Elip label in BTB provide discourse information that is difficult to identify by type (let alone the form) of the missing element(s). These difficult cases can be processed only manually. Usually the additional information could be recovered within the whole text or on the basis of general knowledge. In this case within UD we could use *orphan* relation, but then the encoded information would be lost. In order to preserve this information, we modify the *orphan* relation in order to specify the value the discourse information. For example, *orphan:cop* is used to represent the case of an elided copula licensed by discourse.

If we put the comparison on a broader scale of approaches and thus — beyond enhanced dependencies, then the following observations are to be made. First of all, the idea of using null elements instead of verbs or verbal groups does not cover all other cases with elided elements in UD. For example, in the nominal ellipsis the elided head is substituted by its dependent (if such a dependant is present). This means that a process of substantivization is performed. A similar promotion strategy holds for auxiliaries. In BTB in such cases null elements have been used (recall examples 2 and 4).

In the case of BTB, the process of substantivization is restricted to: a) adjectives promoted to nouns; b) numerals in the structure 'one of them; three of them', etc. In general, the annotation scheme puts a preference to constituent coordination instead of introducing ellipsis. Constituent coordination, however, applies in cases when the coordinated heads have the same selectional criteria or when dependants are selected in the same way by the same head.

There are two special cases of ellipses in BTB which require more attention. The first case is illustrated in Fig. 1 — the `pro-ss` element could be encoded in two ways in UD: (1) via a new null node for the subject; or (2) express it via enhanced dependencies. The second is illustrated in Fig. 2 where the second clause contains an explicit marker for the place of the ellipsis (a dash). In UD this dash could be used functionally instead of introducing a null node.

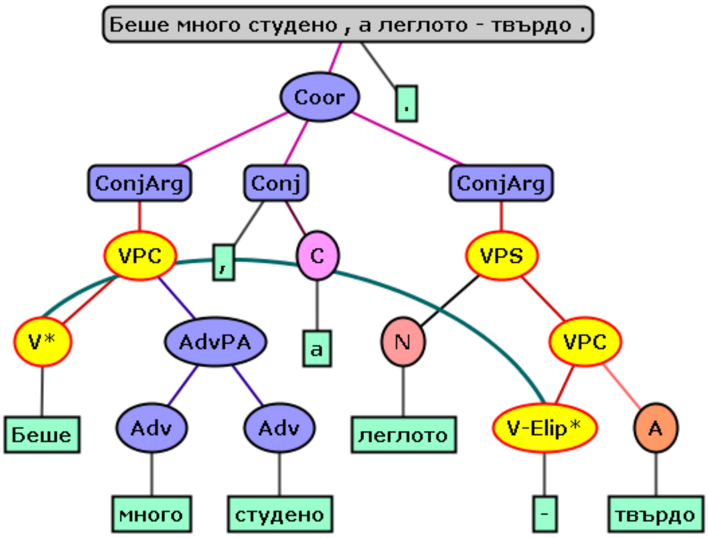


Figure 2: An example of ellipsis marked as a dash in the sentence. Беше много студено, а леглото - твърдо. “Beshe mnogo studeno, a legloto - tvyrdo.” (It was very cold, and the bed - firm.)

5 Conclusions

The current general principles behind UD for handling ellipsis are as follows: a) elided element with no dependents is not processed at all; b) if it has dependants, then they are promoted as heads and c) the promoted element uses the relation *orphan* when other functional elements are attached to it. In BTB, besides the systematically applied null-node-insertion-strategy, ellipsis subtypes were added as a specification relation. Substantivation was kept for the lexicalized in the dictionary dependants.

One possible direction of the UD development would be to extend the null node introduction. Another one is to continue with the mixed strategy of treating ellipses in the basic and enhanced dependencies as it is now.

From our point of view, in both cases it would be useful to add more information on the ellipsis type and characteristics, and also to consider language specific features as it was done for other phenomena. For example, in Bulgarian it is not typical to elide the main predicate and to leave the auxiliary/modal to be promoted as it is in English (e.g. *She wants to go there, but he does not* or *She wanted to go there, but he did not want to*).

As discussed in the cited literature here (and beyond it), the proper treatment of ellipsis in an explicit way is important for the mono- and cross-lingual as well as for reasonable typological surveys across languages.

6 Acknowledgements

This research was partially funded by *the Bulgarian National Interdisciplinary Research e-Infrastructure for Resources and Technologies in favor of the Bulgarian Language and Cultural Heritage, part of the EU infrastructures CLARIN and DARIAH – CLaDA-BG*, Grant number DO01-164/28.08.2018 and the Bulgarian National Science Fund, Grant number 02/12/2016 — *Deep Models of Semantic Knowledge (DemoSem)*. We would like to thank all reviewers for their valuable feedback. All errors remain our own.

References

Kira Droganova and Daniel Zeman. 2017. *Gapping Constructions in Universal Dependencies v2*, Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017), pp. 48–57. Gothenburg, Sweden.

Tomáš Jelínek, Vladimír Petkevic, Alexandr Rosen, Hana Skoumalová and Premysl Vítovec. 2015. *Taking Care of Orphans: Ellipsis in Dependency and Constituency-Based Treebanks*, Proceedings of the TLT14, pp. 119–133. Warsaw, Polland.

Marie Mikulova. 2014. *Semantic Representation of Ellipsis in the Prague Dependency Treebanks*, Proceedings of ROCLING 2014, pp. 125—137. Prague, Czech Republic.

Timothy Osborne and Junying Liang. 2014. *A Survey of Ellipsis in Chinese*,Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015), pp. 271—280. Uppsala, Sweden.

Petya Osenova and Kiril Simov. 2017. *Recent Developments within BulTreeBank*, Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories, pp. 129—137. Prague, Czech Republic.

Adam Przepiorkowski and Agnieszka Patejuk. 2019. *From Lexical Functional Grammar to enhanced Universal Dependencies*, Languages Resources and Evaluation, published online: 04 February 2019. Springer.

Sebastian Schuster, Matthew Lamm and Christopher D. Manning. 2017. *Gapping Constructions in Universal Dependencies v2*, Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017), pp. 123–132. Gothenburg, Sweden.

Kiril Simov, Petya Osenova, Alexander Simov and Milen Kouylekov. 2005. *Design and Implementation of the Bulgarian HPSG-based Treebank*, ournal of Research on Language and Computation. Special Issue, pp. 495–522. Kluwer Academic Publisher.

Rediscovering Greenberg’s Word Order Universals in UD

Kim Gerdes
LPP (CNRS)
Sorbonne Nouvelle, France
kim@gerdes.fr

Sylvain Kahane
Modyco (CNRS)
Université Paris Nanterre, France
sylvain@kahane.fr

Xinying Chen
University of Ostrava, Czech Republic
Xi’an Jiaotong University, China
xy@yuyanxue.net

Abstract

This paper discusses an empirical refoundation of selected Greenbergian word order universals based on a data analysis of the Universal Dependencies project. The nature of the data we work on allows us to extract rich details for testing well-known typological universals and constitutes therefore a valuable basis for validating Greenberg’s universals. Our results show that we can refine some Greenbergian universals in a more empirical and accurate way by means of a data-driven typological analysis.

1 Introduction

Modern research in the field of language typology (Croft 2002; Song 2001), mostly based on Greenberg (1963), focuses less on lexical similarity and relies rather on various structural linguistic indices for language classification and generally puts much emphasis on the syntactic word order of some grammatical relations in a sentence (Haspelmath et al. 2005). Considered as the founder of word order typology, Greenberg (1963) proposed 45 linguistic universals and 28 of them refer to the relative position of syntactic units, such as the linear relative order of subject, object, and verb in a sentence. A more empirical way of examining word order typologies, testing correlations between two binary grammatical relations such as OV vs. VO and SV vs. VS, can be found in Dryer (1992) (following Lehmann 1973), in which, some detailed word order correlations based on a sample of 625 languages are reported.

It is noteworthy that the field of word order typology has a strong empirical tradition, working with data and trying to describe the data with great precision. From a perspective of data analysis, new language data is emerging every day in this so-called era of ‘big data’. It has never been a better moment than today to challenge, test, and corroborate existing ideas based on better and bigger data.

With the appearance of larger sets of treebanks, research has begun to test existing word order typology claims or hypothesis based on treebank data. Investigating treebanks of 20 languages, Liu (2010) tested the ‘traditional’ typological claims with the subject-verb, object-verb and adjective-noun data extracted from the treebanks, with coherent results, also showing that these 20 languages can be arranged on a continuum with absolute head-initial and head-final patterns at the two ends. Liu further states that treebank based methods will be able to provide more complete and fine-grained typological analyses, while previous methods usually had to settle for a focus on basic word order phenomena (Hawkins 1983, Mithun 1987). These new resources allow reviewing and verifying well-known typological claims based on annotations of authentic texts (Liu et al. 2009, Liu 2010, Futrell et al. 2015). [1]

The Universal Dependencies project (UD, Nivre et al. 2016), the basis of the present study, has seen a rapid growth into its present ample size with more than 140 treebanks of about 85 different lan-

[1] The development of treebanks is a cumbersome work. Even 75 languages only cover a modest segment of the world’s languages. Another direction investigated in Östling (2015) is the use of parallel texts as the available translations of the New Testament in 986 languages. Such methods are not the subject of our paper but it is worth considering them for future works, knowing that translations contain some bias and are not fully representative of the target language (especially when the source text belongs to a marked genre such as religious texts).

guages. UD has been developed with the goal of facilitating multilingual parser development, cross-lingual learning, and parsing research from a perspective of language typology (Croft et al. 2017). The annotation scheme is an attempt to unify previous dependency treebank developments based on an evolution of (universal) Stanford dependencies (de Marneffe et al., 2014), Google universal part-of-speech tags (Petrov et al., 2011), and the Interset interlingua for morphosyntactic tagsets (Zeman, 2008). The general philosophy is to provide a universal inventory of categories and guidelines to facilitate consistent annotation of similar constructions across languages, while allowing language-specific extensions when necessary. UD expects the schema, as well as the treebank data, to be "satisfactory on linguistic analysis grounds for individual languages", and at the same time, to be appropriate for linguistic typology, i.e., to provide "a suitable basis for bringing out cross-linguistic parallelism across languages and language families".[2]

One outstanding advantage of using this data set for language typology studies is the sheer size of the data set: we worked on UD 2.2, which includes 110 treebanks in over 70 languages. As all UD treebanks use the same annotation scheme, the database provides rich informative evidence that can be easily compared and interpreted across authentic texts of various languages.

Following Liu (2010), this paper aims to test well-known existing word-order universals based on the data analysis of a set of uniformly annotated texts of diverse languages. Even though the set of languages of UD is currently not well-balanced in terms of language diversity (half of the languages of the database are Indo-European languages and non-Indoeuropean treebanks are often too small to be taken into account for some measures; cf. Bell (1978), Perkins (1989, 2001), Dryer (1989, 1992), Croft (1991), Whaley (1996), Dik (2010) on language sampling) and the results will have to be confirmed in the future on an even wider collection of languages, this resource allows us to have a new take on the question of language universals.

The paper is structured as follows. In Section 2, we introduce dependency treebanks and explain amendments of the current annotation scheme that were necessary to obtain typologically relevant data. In Section 3, we discuss and compare some of Greenberg's (1963) Universals with our results. In the conclusion, we discuss the potential of using UD treebanks for future typological studies.

2 Material and Methods

Dependency trees encode the relations between words by means of an arrow that goes from the head to another element of the phrase (Tesnière 1959 [2015], Mel'čuk 1988). The direction of these arrows, which indicates the relative position of a phrase towards its governor, is the base of our measures. The dependency analysis[3] of the sentence "*Syntactic dependency treebanks help you understand typology*" has three head-initial relations (for example *understand → typology*) and three head-final relations (for example *treebanks ← help*), see Figure 1 for a graphical illustration. Dependency Syntax considers syntactic relations between words independently of word order, and dependency trees can be represented as simple dominance relations. No hypothesis on a basic word order has to be stipulated for the representation itself and the notion of basic word order is foreign to Dependency Syntax: When studying word order in Dependency Syntax, we assess the different linearizations of an unordered dependency tree. Each dependency has two possible linearizations (*governor → dependent* or *dependent ← governor*), one of which may be dominant in the sense that it appears more frequently.

[2] UD introduction page http://universaldependencies.org/introduction.html consulted in August 2017.

[3] The syntactic analysis of this sentence is subject to debate. The proposed analysis corresponds to what is commonly done in dependency syntax. The annotation choices are based on theoretical considerations, for instance the analysis of *you* as an object of *help* rather than as a subject of *understand*. See Hudson (1998) for a comprehensive overview of the stakes of this particular question in a dependency perspective.

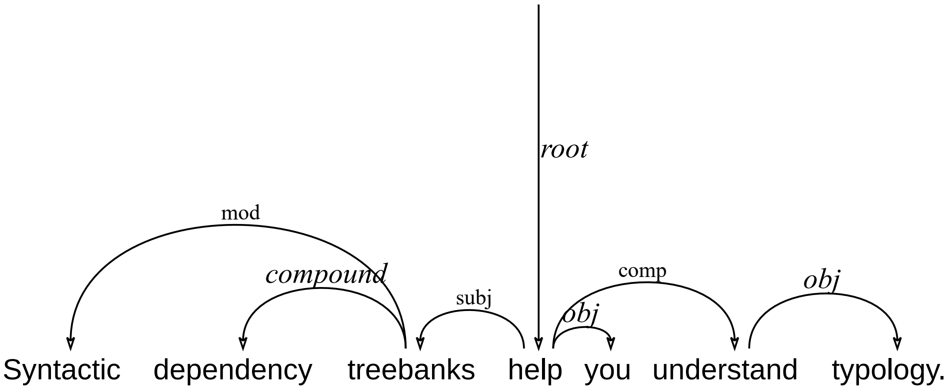


Figure 1: Example of an ordered dependency tree

Our study is based on Surface-Syntactic Universal Dependencies (SUD), a variant of the UD annotation scheme (Gerdes et al. 2018). SUD is better suited for word order studies as it is based on distributional criteria whereas UD favors relations between content words. In SUD, contrary to UD, prepositional phrases are headed by prepositions, and auxiliaries and copula are analyzed just like other matrix verbs, taking the embedded verb as a dependent. The choice of the SUD version is particularly important when we consider a comprehensive view of all constructions of one language, for example Japanese is nearly completely head-final in SUD whereas Japanese UD has a number of head-initial relations such as *adposition-noun* constructions and *auxiliary-verb* constructions.

From these treebanks, we can compute for any relation the percentage of head-initial links. We can also filter the links of any given relation by the POS of the governor or of the dependent to look into more specific sub-cases. For instance, we were interested in a separation of the object relation (*comp:obj* in SUD) into V pronO (*VERB-comp:obj>PRON*) and V nomO (*VERB-comp:obj>NOUN*) (pronominal vs nominal object) and of the subject relation into -*subject>PRON* and -*subject>NOUN* (pronominal vs nominal subject). For each relevant *POS-relation>POS* triple (as well as *POS-relation>*, -*relation>POS*, and *relation*) and each of the UD languages (merging all treebanks of the same language),[4] we computed the number of head-initial and head-final dependencies.

The scatter plot of Figure 2 shows the percentage of head-initial head-daughter dependencies, that is, dependencies that link a head with a constituent that is subordinated to it. We do not consider coor-

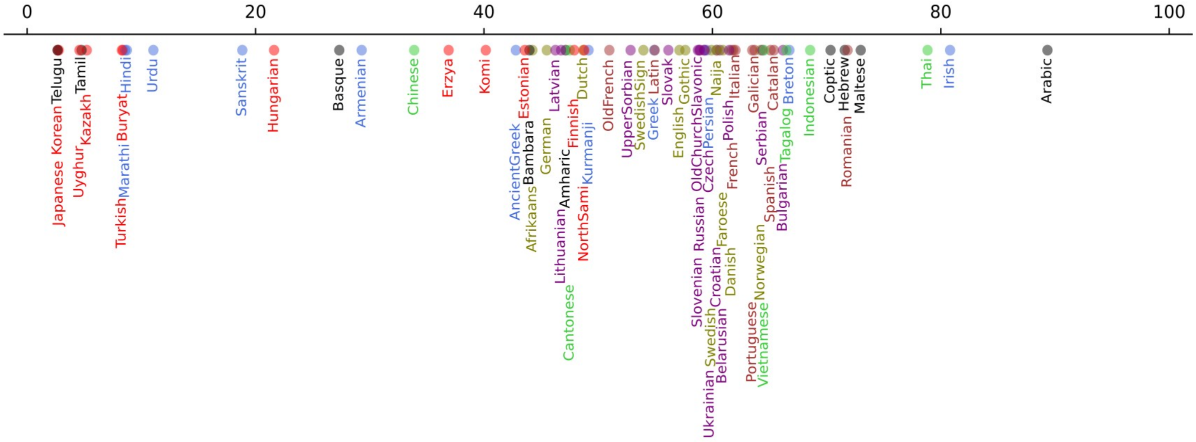


Figure 2: Percentage of head-initial head-daughter dependency relations in the UD treebanks ranging from 3% for Japanese to 89% for Arabic.

[4] We are aware that treebank properties not only reflect the language but also show genre differences as well as annotation choices. As shown in Chen & Gerdes (2017), the global measures for different treebanks of the same language remain nevertheless quite homogeneous.

dination for instance, although coordination can also be encoded with the same formal device of "dependency". For a discussion on the criteria that allows deciding whether a construction is clearly headed (endocentric in the terms of Bloomfield 1933), see for instance Criteria B of Mel'čuk (1988). The list of SUD/UD relations we eliminated includes *conj*, *appos*, *reparandum*, *fixed*, *flat*, *compound*, *list*, *parataxis*, *orphan*, *goeswith*, *punct*, *root*, *dep*, and *clf*. We decided to keep the *det* relation for determiners, even if the relation linking a determiner and a noun does not always provide a clear-cut head (cf. the DP-hypothesis; Hudson 1984, Abney 1987). One of the reasons we keep the relation is that it has been used even in some languages, such as Japanese, which do not have clear determiners, for closed classes of adjectives which have a similar meaning as English determiners. (We consider that a language has clear determiners when the noun cannot be used alone in some argument positions.)

3 Results and Discussion

"*Universal 19. When the general rule is that the descriptive adjective follows, there may be a minority of adjectives which usually precede, but when the general rule is that descriptive adjectives precede, there are no exceptions.*"

This Greenbergian universal means that languages with dominant ADJ-NOUN order (that is, with a dominant head-final *NOUN-dependent>ADJ* relation), must necessarily have a very low percentage of head-initial occurrences. In other words, a gap in the area of moderately head-final languages is expected for this relation.

If we look at the distribution of languages for the *NOUN-dependent>ADJ* relation in Figure 3, we see that Universal 19 is more or less confirmed. On one hand, there is no real gap in the distribution of dominant head-final languages, due to the presence of Polish and Old French between 20% and 50%.[5] On the other hand, we observe that the distribution of head-initial languages is much more uniform than the distribution of head-final languages, whose languages are highly concentrated between 0% and 5%. More precisely, the average percentage of head-initial languages is 83.4% with a standard deviation (SD) of 14.2. On the left side of the graph, we obtain an average of 3.8% and an SD of 9.1, which confirms the universal statistically.[6]

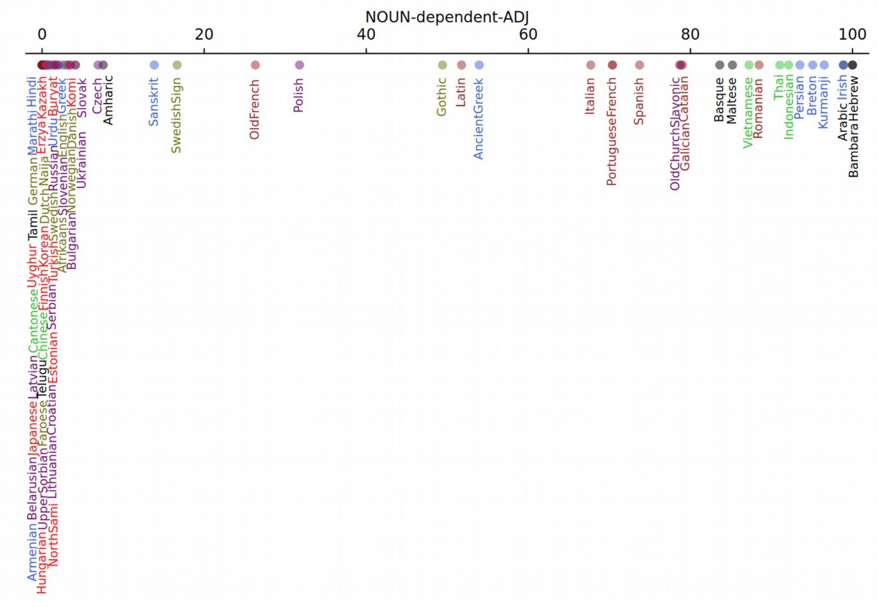


Figure 3: Language distribution for the direction of the NOUN-dependent-ADJ relation.

[5] A possible explanation for the presence of Old French is that the Old French UD treebank covers a wide period (842 to 1225, see Stein & Prévost 2013), where Latin, positioned at around 50% in our diagram, was influenced by Germanic tribes. We have no explanation why Polish is an outlier among the modern Slavic languages.

[6] Let us recall that standard deviation measures the average deviation of the language positions from the mean. In other words, these measures confirm what can be observed on the diagram: The languages on the left of the diagram are more concentrated and very much left-leaning, while the languages on the right are more central and more balanced.

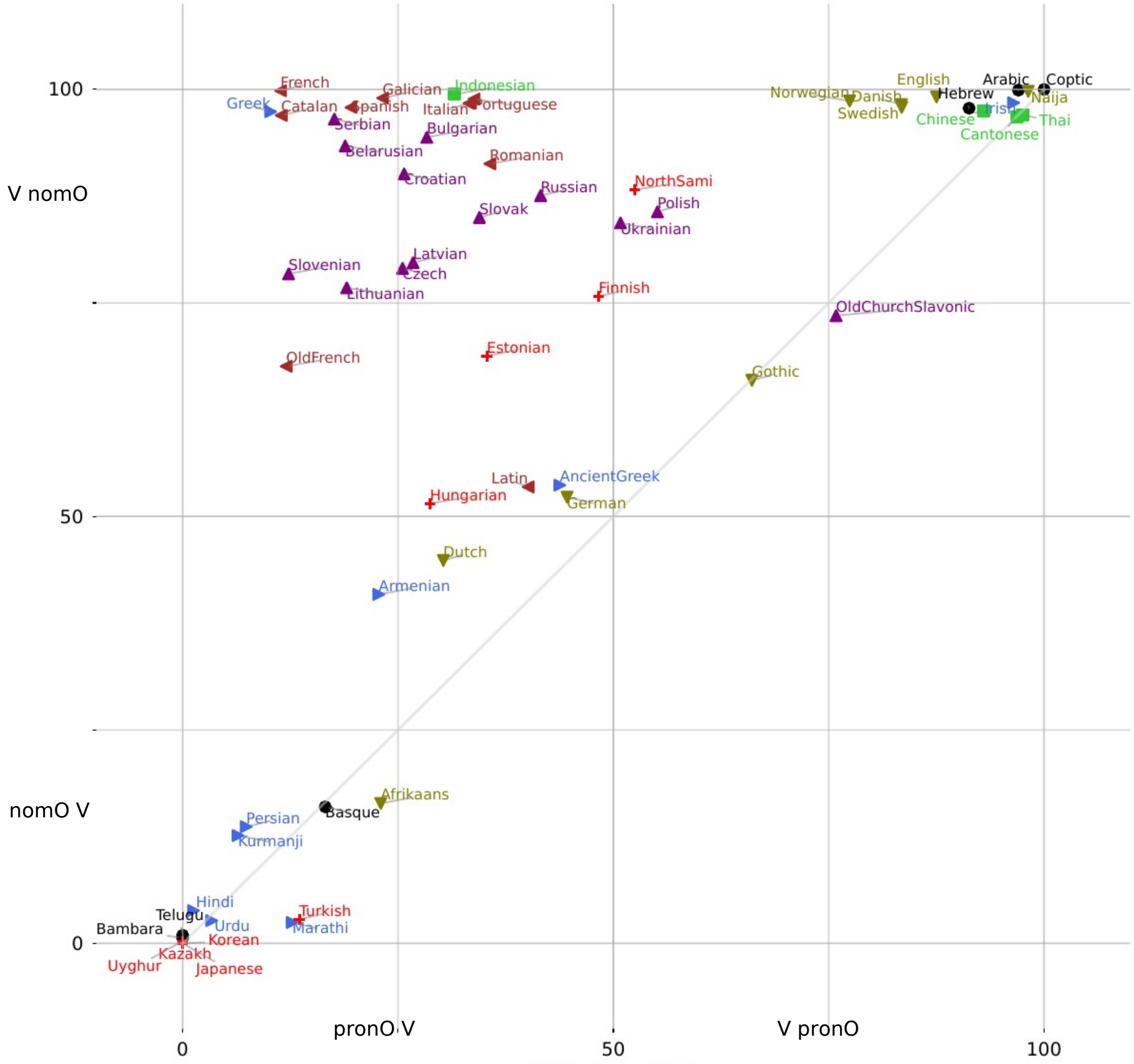


Figure 4: Scatter plot of the percentage of V pronO compared to V nomO

Indo-European languages: triangles: Indo-European-Romance: brown ◀, Indo-European-Baltoslavic: purple ▲ Indo-European-Germanic, including the English Creole Naija: olive ▼, other Indo-European: blue ▶

Sino-Austronesian: green squares ■. Agglutinating languages: red plus signs +. Other languages (Afroasiatic and Dravidian languages as well as Basque): black circles ●

Some language points are hidden because the available treebank data for the language is not sufficient to provide significant measurements; more specifically, we decided to eliminate every language with less than 50 occurrences of one of the two compared types of relations.

When analyzing further the Greenbergian Universal 19, we note that the interpretation of the condition "when the general rule is that the descriptive adjective follows" is difficult to apply empirically. If we take this rule to hold for all languages with predominant NOUN-ADJ order (i.e. with a *NOUN-dependent>ADJ* relation score of more than 50%), we include the classical languages Latin, Gothic, and Ancient Greek in this group although their position is just above 50%. A universal such as Universal 19 tries to describe the distribution of languages considering a special feature (the distribution of *ADJ*s towards the NOUN) in qualitative terms, which is not straightforward. We believe that a diagram such as Figure 3 can be a more satisfying alternative to such descriptions since it provides many more details.

"*Universal 25. If the pronominal object follows the verb, so does the nominal object.*"

Universal 25 is a universal referring to a qualitative absolute property such as the "basic word order" of a language, and not to a numerical threshold. It supposes that we can categorize languages into languages where "the pronominal object follows the verb" and languages where "the pronominal object does not follow the verb", as well as languages where "the nominal object follows the verb" and languages where "the nominal object does not follow the verb". Therefore, Universal 25 is an implicational universal, because it has the form of an implication between two statements: "the pronominal object follows the verb" (V pronO) and "the nominal object follows the verb" (V nomO). Universal 25 can be abbreviated as V pronO → V nomO.

Let us see now how Universal 25 is related with the scatter plots in Figure 4. We can remark that Greenberg's statement is not totally clear. What does it mean that "the pronominal object follows the verb"? Does it mean that pronominal objects always follow the verb or does it mean that in most cases they follow the verb? Is there any quantitative statement hidden in Greenberg's statement? Whatever the answer to these questions might be, we can translate the statements of Universal 25 into more satisfying, quantitative statements and see whether the implication is verified on our data. In other words, "the pronominal object follows the verb" (V pronO) can be interpreted as: "the percentage of pronominal object on the right of the verb is greater that *a*", where *a* is some relevant threshold. For instance, for *a* = 75%, we verify what is a first tentative quantitative universal:

Universal 25': For every language, if the percentage of pronominal objects on the right of the verb is greater than 75%, so is the percentage of nominal objects on the right of the verb.

We abbreviate Universal 25' by: V pronO ≥ 75% → V nomO ≥ 75%. Universal 25' is illustrated by Figure 5a. Let us recall that the negation of a property A → B is A & ¬B. Thus, Universal 25' claims that there is no language with V pronO ≥ 75% and V nomO < 75%, that is, that the corresponding rectangle in Figure 5a (hatched in gray) is empty of any language.

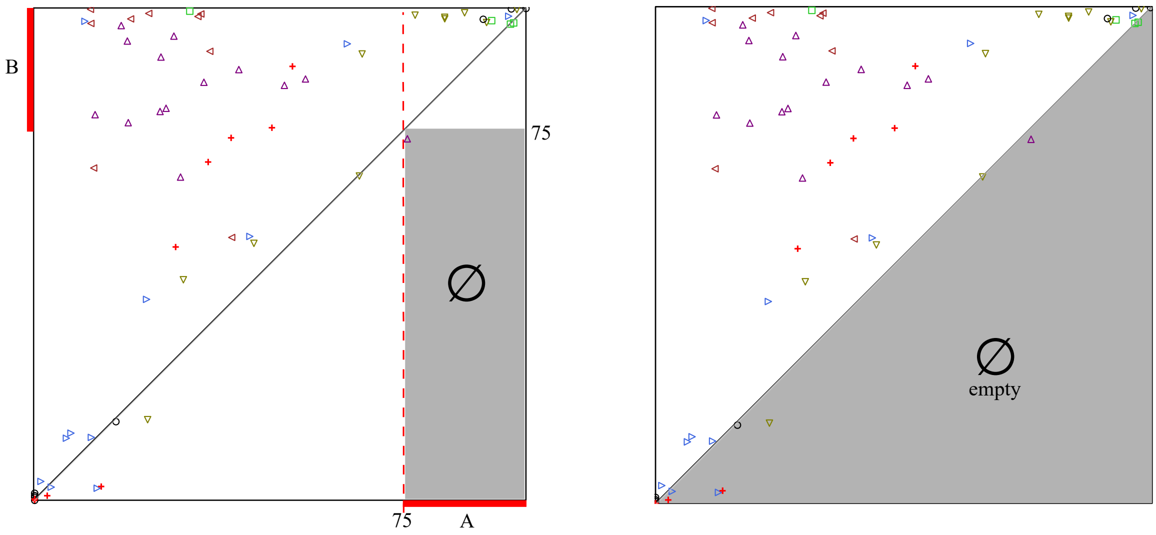


Figure 5: Universal 25'

a. V pronO ≥ 75% → V nomO ≥ 75%
b. V nomO ≥ V pronO (that is, for every a, V pronO ≥ a → V nomO ≥ a)

Yet, we do not know the relevant threshold *a*. If *a* = 100%, Greenberg's universal only concerns languages with very strict order, where all pronominal objects are on the right of the verb. On the other side, if *a* = 50%, it concerns many more languages, that is, all the languages that place more pronominal objects on the right of the verb than on the left. But if the universal concerns more languages, the statement for each of these languages is also less strong, because it only says that these languages place more nominal objects on the right than on the left. We believe that qualitative universals such as Universal 25 (V pronO → V nomO) should be interpreted by means of quantitative universals such as "V pronO ≥ a → V nomO ≥ a" for some a, so that we can obtain more accurate claims for language universals.

Another direction is to not consider a particular threshold at all. For our example, we do not need to propose a threshold, because for almost all languages, we have V pronO ≥ a → V nomO ≥ a for every a, which is equivalent to V nomO ≥ V pronO, which gives us the following universal:

Universal 25": Almost every language has a higher proportion of nominal objects than of pronominal objects on the right of the verb.

This last statement is verified on our data and corresponds to a near empty triangular form in Figure 5b. Universal 25" has no equivalent in terms of qualitative universals à la Greenberg. Thus working with quantitative data opens up the door to completely new universals.

4 Conclusion

Our results roughly confirm Greenberg's word order universals 19 and 25 in that these two universals are coherent with the empirical analysis based on the treebanks of more than 70 languages in UD. However, we also can see obvious limitations of Greenberg's universals in our discussion. To be more specific, Greenberg's universals remain to a certain extent vague, since they are purely implicational, and should be updated into a more accurate and empirically verifiable description, going along with the growing treebank data resources and computing power that are available our days.

In this pilot study, we present one way of accomplishing this task. Commonly, typological universals declare or can be interpreted as the impossibility (or statistical rareness) of languages with certain properties. As we have shown in our study, some of Greenberg's universals about word order have this type of configurational interpretation. By introducing more informative quantitative descriptions with broader conditions, we can establish more sophisticated quantitative universals which provide more accurate descriptions and actually can generalize Greenberg's universals. For example, Universal 25 is in fact (almost) true for every a, giving us a triangular pattern, which paves the way for other types of universals, where we would actually describe universal restrictions on human languages as the shapes that the clouds of language take on scatterplots of various properties.

Reference

Abney, S. P. (1987). *The English noun phrase in its sentential aspect*, Doctoral dissertation, Cambridge: MIT.

Bloomfield, L. (1933). *Language*. New York: Henry Holt.

Bell, A. (1978). Language samples. In J. H. Greenberg, C. A. Ferguson, E. A. Moravcsik (eds.), *Universal off Human Languages, Vol. I: Method-Theory, 123-156.*

Chen, X., K. Gerdes (2017). Classifying Languages by Dependency Structure. Typologies of Delexicalized Universal Dependency Treebanks, *Proceedings of the conference on Dependency Linguistics* (DepLing).

Croft, W. (1991). *Syntactic categories and grammatical relations: The cognitive organization of information*. University of Chicago Press.

Croft, W. (2002). *Typology and universals*. Cambridge University Press.

Croft, W., D. Nordquist, K. Looney, M. Regan (2017) Linguistic Typology meets Universal Dependencies. *Proceedings of the conference on Treebanks and Linguistic Theorie*s (TLT), 63-75.

De Marneffe, M.-C., T. Dozat, N. Silveira, K. Haverinen, F. Ginter, J. Nivre, C. D. Manning (2014). Universal Stanford dependencies: A cross-linguistic typology. *Proceedings of LREC*. Vol. 14.

Dik, B. (2010). Language Sampling, in Song, J.J. (ed) *The Oxford Handbook of Linguistic Typology*. Oxford Handbooks.

Dryer, M. S. (1989). Large linguistic areas and language sampling. *Studies in language*, 13(2), 257-292.

Dryer, M. S. (1992). The Greenbergian word order correlations. *Language*, 68, 81-138.

Futrell, R., K. Mahowald, E. Gibson (2015). Quantifying Word Order Freedom in Dependency Corpora, *Proceedings of the conference on Dependency Linguistics* (*DepLing*).

Gerdes, K., B. Guillaume, S. Kahane, G. Perrier. (2018). SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. *Proceedings of Universal Dependencies Worksho*p.

Greenberg, J. H. (1963) Some universals of grammar with particular reference to the order of meaningful elements. In J. H. Greenberg (ed.) *Universals of grammar*, Cambridge: MIT, 73-113.

Haspelmath, M., M. S. Dryer, D. Gil, B. Comrie (2005). *The World Atlas of Language Structures Online*. Munich: Max Planck Digital Library.

Hawkins, J. A. (1983). *Word order universals: Quantitative analyses of linguistic structure*. New York: Academic Press.

Hudson, R. (1984). *Word Grammar*. Oxford: Basil Blackwell.

Hudson, R. (1998) Functional control with and without structure-sharing. *Typological studies in language*, 38, 151-170.

Lehmann, W. P. (1973) A Structural Principle of Language and its Implications. *Language*, 49, 47-66.

Liu, H. (2010). Dependency direction as a means of word-order typology: A method based on dependency treebanks. *Lingua*, 120(6), 1567-1578.

Liu, H., Y. Zhao, W. Li (2009). Chinese syntactic and typological properties based on dependency syntactic treebanks. *Poznań Studies in Contemporary Linguistic*s, 45(4), 509-523.

Mel'čuk, I. A. (1988). *Dependency syntax: theory and practice*. New York: SUNY press.

Mithun, M. (1987) Is basic word order universal?. In R. Tomlin (ed.) *Grounding and Coherence in Discourse*. [Typological Studies in Language, 11], Amsterdam: John Benjamins. 281-328. Reprinted in D. Payne (ed.) (1992). *The Pragmatics of Word-Order Flexibility* [Typological Studies in Language, 22], Amsterdam: John Benjamins. 15-61.

Nivre, J., M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. T. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty (2016). Universal Dependencies v1: A Multilingual Treebank Collection. *Proceedings of LREC.*

Östling, R. (2015) *Bayesian Models for Multilingual Word Alignmen*t, Doctoral dissertation, Stockholm University.

Petrov, S., D. Das, R. McDonald (2011). A universal part-of-speech tagset. *arXiv preprint*, arXiv:1104.2086.

Perkins, R. D. (1989), Statistical techniques for determining language sample size. *Studies in Language*,13(2), 293-315.

Perkins, R. D. (2001). Sampling procedures and statistical methods. In M. Haspelmath, E. König, W. Oesterreicher, W. Raible (eds.). *Language Typology and Language Universals: An International Handbook*. Vol. 1. Berlin: De Gruyter, 419-434.

Song, J. J. (2001) *Linguistic Typology: Morphology and Syntax*. Pearson Education.

Stein, A., S. Prévost (2013). Syntactic annotation of medieval texts: the Syntactic Reference Corpus of Medieval French (SRCMF). In P. Bennett, M. Durrell, S. Scheible, R. Whitt (eds) *New Methods in Historical Corpus Linguistics, Corpus Linguistics and International Perspectives on Language*, CLIP Vol. 3. Tübingen: Narr., 75-82.

Tesnière, L. (1959). *Eléments de syntaxe structurale*. Paris: Klincksieck. [Transl. by T. Osborne T., S. Kahane (2015) Elements of structural syntax. Benjamins].

Whaley, L. J. (1996) *Introduction to typology: the unity and diversity of language*. Sage Publications.

Zeman, D. (2008) Reusable Tagset Conversion Using Tagset Drivers. *Proceedings of LREC.*

Building minority dependency treebanks, dictionaries and computational grammars at the same time—an experiment in Karelian treebanking

Tommi A Pirinen
Universität Hamburg
Hamburger Zentrum für Sprachkorpora
Max-Brauer-Allee 60, D-22765 Hamburg
tommi.antero.pirinen@uni-hamburg.de

Abstract

Building a treebank from scratch can easily be an elaborate, highly time consuming task, especially when working with a minority language with moderately complex morphology and no existing resources. It is also then typically true that language experts and informants with suitable skill sets are a very scarce resource. In this experiment I have attempted to work in parallel on building NLP resources while gathering and annotating the treebank. In particular, I aim to build a decent coverage morphologically annotated lexicon suitable for rule-based morphological analysis as well as accompanying rules for basic morphosyntactic analysis. I propose here a workflow, that I have found useful in avoiding redoing same work with related NLP resource construction.

1 Introduction

Karelian languages are languages closely related to Finnish spoken mainly in the republic of Karelia in Russia and surroundings. The languages are split in the ISO 639–3 standard between a few language codes: *Karelian* (krl) and *Livvi* or Olonets karelian (olo) for the two main branches of the language. The fact that 'krl' is commonly refered to as just Karelian can be confusing because 'olo' is also Karelian but I try to make the distinction clear throughout the article by using the ISO codes when necessary. The division is not totally unproblematic but I have followed it in the treebank for ease of development and use. There are some 35,000 native speakers of Karelian (krl) [1] and 31,000 for Livvi (olo) [2] according to Ethnologue, and both are classified as "Developing". The languages are developed enough to have some grammars (Zaikov, 2013; Ahtia, 1938; Markianova, 2002), dictionaries and books written, as well as some regular newspapers and broadcasts, but very few digital or computational resources so far. For unannotated corpora I have found a source with freely usable texts classified according to ISO language codes.

This paper discusses creation and ongoing work for two Karelian treebanks and compatible morphological parsers. The first part of the Karelian data will be included in the 2.4 release of the Universal Dependencies and I hope to enlarge and verify the data with native informants as well as include the Livvi data by the next release. The treebanks were named under the abbreviation of KKPP or *Karjalan kielten puupankit* which is Finnish for Karelian treebanks.

The rest of the article is organised as follows: in Section 2 I describe the languages and our goals for the treebanking, in Section 3 I describe the tools and methods for building treebanks, in Section 4 I describe the corpus selection and finally in Section 5 I summarise the article and talk about future work and ideas.

2 Background

As languages with very few available NLP resources, one of our first goals is to get annotated corpora. The universal dependencies format is a good choice for a standard for writing a new treebank at the moment; it has been used with many Uralic languages already that provide for reference for difficult

[1] https://www.ethnologue.com/18/language/krl/

[2] https://www.ethnologue.com/18/language/olo/

situations. Also, the North Saami treebank was made based on a rule-based finite-state morphological analyser (Sheyanova and Tyers, 2017), building one of which is also a goal for us, so I can safely say that the two formats are compatible and complement each other. One of the reasons why I make morphological analysers is to be able to provide number of end-user tools like spell-checking and correction as well as the reference corpus, for example in other Uralic languages there are plenty of resources hosted by giellatekno (Moshagen et al., 2014).

When I started with the treebanking, morphological analyser writing task, there were virtually no freely available corpora for Karelian and also no electronical dictionaries or analysers for Karelian krl. There was an existing analyser for Livvi and for that reason I have started our project with Karelian first. For digitised paper dictionaries, I have a dictionary for Karelian languages[3], that covers both Karelian and Livvi. The overall format and transcription differences, however, make it not directly usable for a source dictionary for morphological analyser for Karelian languages but rather an semi-automated source reference.

One of the thing I have established in the research of under-resourced languages in Uralic space is that for the survival and digital survival of a language certain technological resources need to be developed, and our aim with this project is to build as many of the necessary resources rapidly as possible.

One of the things that I have taken into consideration working on this treebank is how corpora are built within Uralic linguistic community outside the Universal Dependencies, e.g. in documentary linguistics. One of the prominent paradigms there is based on the line of tools from SIL shoebox to Fieldworks Explorer (FLeX), the workflow within those makes use of building corpora and dictionary simultaneously and this experiment is in a way our precursory study to implementing a similar tool for dependency treebanking style of linguistics. For reference on such Uralic research within computational linguistics see (Blokland et al., 2015).

Furthermore I are developing a morpho-syntactic rule-based methodology that can provide partial, ambiguous dependency graphs. The approach of building rule-based analysers first is very prominent within computational linguistics research of Uralic languages. In this article I are aiming to connect the traditional development of rule-based morphological analysers into treebanking workflow in a manner that optimises the usage of native informants' and the computational linguists' time, which is a crucial component for development in a very under-resourced setting.

Finally, I aim to have wide coverage of Uralic languages in the Universal Dependency project treebanks, and further study and experiment in the state-of-the-art methodology in large variety of NLP and typological research topics that have been empowered by the project. At the moment there are 6 Uralic treebanks available: Finnish (Haverinen et al., 2014; Voutilainen et al., 2012), Estonian (Muischnek et al., 2016), Hungarian (Vincze et al., 2010), North Saami (Sheyanova and Tyers, 2017), Komi (Partanen et al., 2018), and Erzya (Rueter and Tyers, 2018), out of some 30 that can easily have treebanks.

3 Methods

One of the contributions of this article is, that I am developing a sustainable workflow for creation of a wide array of technological resources for a seriously under-resourced language. For language technology infrastructure I will make use of an existing language technology infrastructure developed by (Moshagen et al., 2014), which I have selected because it provides a number of necessary components for free once morphological analysers are built, e.g. automatic spell-checking, machine-translation and so on.

The morphological analysis is based on the finite-state morphology (Beesley and Karttunen, 2003), this means in practice that one needs to build a dictionary and morphological rules describing the morphological processes. To couple the dictionary building with treebanking effort I have developed a method to generate lexicon entries from the annotated treebank data. I also use the analysers to generate suggestions for the annotators for the dependency annotations.

To give an example of the resource building workflow, a sentence might be annotated in CONLL-U format like:

```
# sent_id = vepkar-1774.7
```

[3] http://kaino.kotus.fi/cgi-bin/kks/kks_etusivu.cgi

```
# text = - Myö toivomma, jotta meijän kuččuh vaššatah starinankertojat ta guslinšoittajat, jotta kaččojat šuahah nähä
vanhanaikasien rahvahantapojen rekonstruointie, koroššetah järještäjät..
1       -       -       PUNCT   PUNCT   _       3       punct   _       Weight=0.0033333333333333335
2       Myö     myö     PRON    PRON    Case=Nom|Number=Sing|Person=1|PronType=Prs      3       nsubj   _       Weight
=500.0
3       toivomma        toivuo  VERB    VERB    Mood=Ind|Number=Plur|Person=1|Tense=Pres|VerbForm=Fin|Voice=Act 0
       root    _       Weight=0.0194|SpaceAfter=No
4       ,       ,       PUNCT   PUNCT   _       8       punct   _       Weight=518.6755555555555
5       jotta   jotta   SCONJ   SCONJ   _       8       mark    _       Weight=0.002142857142857143
6       meijän  myö     PRON    PRON    Case=Gen|Number=Plur|Person=1|PronType=Prs      7       nmod:poss       _
       Weight=500.0
7       kuččuh  kučču   NOUN    NOUN    Case=Ill|Number=Sing    8       obl     _       Weight=500.0
8       vaššatah        vaššata VERB    VERB    Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act 3
       ccomp   _       Weight=500.0248
9       starinankertojat        starinan#kertoja        NOUN    NOUN Case=Nom|Number=Plur    15      nsubj   _  _
```

For a rule-based morphological parser an entry is needed to have at least dictionary form or lemma, and a paradigm for inflectional information; for languages like Karelian one cannot fully guess an entry for an inflectional paradigm from a single example but can usually give quite short list of plausible choices. So, I always extend our dictionaries with the entries from the annotated trees.

Likewise when annotating, I use the morphological analyser that is readily built with UD analyses: lemmas, UPOS and morphological features as well as some rough guesses when possible for the deps (e.g. puncts, Case-based dependencies); the python-based guesser for dependencies can currently handle things like: select `PUNCT` and suggest an attachment to each of the `VERB`s in sentence with `punct` dep, or select feature `Case=Acc` and suggest attachment to all `VerbForm=Fin` in the sentence with an `obj` dep. Thus, I can generate suggestion lists like:

```
# sent-id: <stdin>.21
# text: Koštamukšelaiset toivotah, jotta Koštamukšen ta Petroskoin šekä muijen
# kaupunkien välillä olis järješšetty šiännöllini lentoyhteyš.
1  Koštamukšelaiset  Koštamukšelaiset  X  X  _  _  _  _  SpaceBefore=No|_
2  toivotah  toivuo  VERB  VERBMood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act  0  root  _ _
2  toivotah  toivuo  VERB  VERB  Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Pass  0  root  _  SpaceAfter=No
3  ,  ,  SYM  SYM  _  _  _  _  SpaceBefore=No|Weight=506.4
3  ,  ,  PUNCT  PUNCT  _  2  punct  _  SpaceBefore=No|Weight=0.0033333333333333335
3  ,  ,  PUNCT  PUNCT  _  13  punct  _  SpaceBefore=No|Weight=0.03333333333333333
4  jotta  jotta  SCONJ  SCONJ  _  13  mark  _  Weight=0.0225
5  Koštamukšen  Koštamukšen  X  X  _  _  _  _  _
6  ta  ta  CCONJ  CCONJ  _  7  cc  _  Weight=0.01
7  Petroskoin  Petroskoi  PROPN  PROPN  Case=Gen|Number=Sing  2  obj  _  PropnType=Top|Weight=0.016666666666666666
```

A linguist is provided with this suggestion list per token in order defined by the weights,at the moment expert-determined rule-weighting but when we have large enough corpus I can easily incorporate the unigram log probabilities into weights as well. It should be noted that the linguist is allowed to discard all suggestions and this shall not be considered an unusual case while simultaneously building the analyser and the treebank. The current annotators also use an editor that is automatically running the validation tests[4] for UD after each edit and highlighting problems on the fly. The tools that I have developed so far will also be released with a free/libre open source licence.

When working on the annotation and guidelines I relied quite heavily on existing Uralic treebanks, especially Finnish since it is a closely related language with three treebanks and documentation. For many structures it is possible to find near or exact match using treebank search [5]. For example, the copula structure including the possession structure is marked in the same way in Finnish and Karelian languages, and generally many cases, function words and so forth, overlap with few systematic changes (e.g. in most parts of Karelian (krl) adessive and ablative have same form). Many of the examples where I did not find equivalents in Finnish I looked at other Uralic languages, or Russian, for example in elliptical structures a long hyphen is often used in Karelian and Russian to mark some elided tokens but not in contemporary Finnish in the genres of the UD treebanks at least.

Finally, this workflow goes on to ensure that the morphological analysers I build will have virtually a 100 % coverage of the treebank released, with a very high rate of recall for the treebank fields: lemma, UPOS and the lexical and morphological feature definitions. The reason recall is not 100 % is that there will be some annotations that, while theoretically correct, are not wanted in a normative analyser, e.g. colloquial uses of certain case forms in a role that is not the literary standard, as well as typos and mistakes,

[4] https://github.com/universaldependencies/tools/validate.py

[5] http://bionlp-www.utu.fi/dep_search/

Language	Lexicon size
Karelian	1452
Livvi	56,377

Table 1: The sizes of analysers of Uralic languages.

Treebank	Dependency trees	Syntactic words
Karelian	228	3094
Livvi	20	461
Finnish	34,859	377,822
Estonian	32,385	461,531
North Saami	3122	26,845
Hungarian	1800	42,032
Erzya	1550	15,790
Komi	307	3304

Table 2: The sizes of treebanks of Uralic languages. Dependency trees is number of annotated sentences and syntactic words as defined in UD guidelines.

however, I might change this practice in the future with universal feature `Style=Coll`.[6]

4 Data

There is not a great amount of available data written in Karelian languages to begin with. Furthermore, while there have been written texts for some time, the newest standard ortographies are quite recent, and there is some amount of variation from text to text in the written forms that is not the same as with older more standardised languages. Added to that is that telling languages apart, especially in less standard more dialectal writing, becomes non-trivial task. I started my data collection with web-crawling, and eventually found a corpus collection web site with open licencing policy, and the languages I want to work on categorised by language and genre, called VepKar.[7] The open licence also lets us work on articles instead of shuffled sentences, so it is another advantage.

By the time of writing I have developed a releasable treebank for Karelian and a morphological analyser, which are summarised in the table 2, I have also begun the work on Livvi treebank, which already had a usable analyser in place. For comparison I show some of the other existing Uralic treebanks for reference. Number of dependency trees annotated for non-Karelian languages is based on universaldependencies.org's statistics.

5 Discussion and future work

I have achieved a baseline universal dependency treebank and a morphological analyser for a minority language without pre-existing resources, and started working on a second treebank on a language with pre-existing analyser. In the next part I will contact more experts to verify the analyses and work on extending the treebanks as well as the analysers.

6 Acknowledgments

The author was employed by CLARIN-D during the project.

[6] I thank the anonymous reviewer for the helpful suggestion.

[7] `http://dictorpus.krc.karelia.ru/`

References

Edvard Vilhelm Ahtia. 1938. *Karjalan kielioppi*. Karjalan Kansalaisseura.

Kenneth R Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI publications.

Rogier Blokland, Marina Fedina, Ciprian Gerstenberger, Niko Partanen, Michael Rießler, and Joshua Wilbur. 2015. Language documentation meets language technology. In *Septentrio Conference Series*, number 2, pages 8–18.

Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. 2014. Building the essential resources for finnish: the turku dependency treebank. *Language Resources and Evaluation*, 48(3):493–531.

Ludmila Markianova. 2002. Karjalan kielioppi 5-9. *Periodika, Petroskoi*.

Sjur Moshagen, Jack Rueter, Tommi Pirinen, Trond Trosterud, and Francis M Tyers. 2014. Open-source infrastructures for collaborative work on under-resourced languages. In *Proceedings of e Ninth International Conference on Language Resources and Evaluation, LREC*, pages 71–77.

Kadri Muischnek, Kaili Müürisep, and Tiina Puolakainen. 2016. Estonian dependency treebank: from constraint grammar tagset to universal dependencies. In *LREC*.

Niko Partanen, Rogier Blokland, KyungTae Lim, Thierry Poibeau, and Michael Rießler. 2018. The first komi-zyrian universal dependencies treebanks. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 126–132.

Jack Rueter and Francis Tyers. 2018. Towards an open-source universal-dependency treebank for erzya. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 106–118.

Mariya Sheyanova and Francis M. Tyers. 2017. Annotation schemes in north sámi dependency parsing. In *Proceedings of the 3rd International Workshop for Computational Linguistics of Uralic Languages*, pages 66–75.

Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010. Hungarian dependency treebank.

Atro Voutilainen, Kristiina Muhonen, Tanja Katariina Purtonen, Krister Lindén, et al. 2012. Specifying treebanks, outsourcing parsebanks: Finntreebank 3. In *Proceedings of LREC 2012 8th ELRA Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA).

Pekka Zaikov. 2013. *Vienankarjalan kielioppi*.