

TLT 2019

**18th International Workshop on
Treebanks and Linguistic Theories
(TLT, SyntaxFest 2019)**

Proceedings

28–29 August, 2019
held within the **SyntaxFest** 2019, 26–30 August
Paris, France

©2019 The Association for Computational Linguistics

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

ISBN 978-1-950737-64-2

Preface

The 18th edition of the International Workshop on Treebanks and Linguistics (TLT) follows an annual series that started in 2002, in Sozopol, Bulgaria. TLT addresses all aspects of treebank design, development, and use, "treebank" taken in a broad sense of any spoken or written data augmented with computationally processable annotations of linguistic structure at various levels.

This year's edition is special as TLT is 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 led 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)
Çağrı 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 Artificial Intelligence)
Relja Vulcanovic (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

SyntaxFest 2019 Invited talk - Quantitative Computational Syntax: dependencies, intervention effects and word embeddings	1
<i>Paola Merlo</i>	
Are formal restrictions on crossing dependencies epiphenominal?	2
<i>Himanshu Yadav, Samar Husain and Richard Futrell</i>	
A Surface-Syntactic UD Treebank for Naija	13
<i>Bernard Caron, Marine Courtin, Kim Gerdes and Sylvain Kahane</i>	
Can Greenbergian universals be induced from language networks?	25
<i>Kartik Sharma, Kaivalya Swami, Aditya Shete and Samar Husain</i>	
Parallel Dependency Treebank Annotated with Interlinked Verbal Synonym Classes and Roles	38
<i>Zdeňka Urešová, Eva Fučíková, Eva Hajičová and Jan Hajič</i>	
Ordering of Adverbials of Time and Place in Grammars and in an Annotated English-Czech Parallel Corpus	51
<i>Eva Hajičová, Jiří Mírovský and Kateřina Rysová</i>	
Weighted posets: Learning surface order from dependency trees	61
<i>William Dyer</i>	
Linked Open Treebanks. Interlinking Syntactically Annotated Corpora in the XXX Knowledge Base of Linguistic Resources for Latin	74
<i>Francesco Mambrini and Marco Passarotti</i>	
Challenges of Annotating a Code-Switching Treebank	82
<i>Özlem Çetinoğlu and ÇağrıÇöltekin</i>	
Dependency Parser for Bengali-English Code-Mixed Data enhanced with a Synthetic Treebank	91
<i>Urmi Ghosh, Dipti Sharma and Simran Khanuja</i>	
tweeDe –A Universal Dependencies treebank for German tweets	100
<i>Ines Rehbein, Josef Ruppenhofer and Bich-Ngoc Do</i>	
Creating, Enriching and Valorizing Treebanks of Ancient Greek	109
<i>Alek Keersmaekers, Wouter Mercelis, Colin Swaelens and Toon Van Hal</i>	
Syntax is clearer on the other side - Using parallel corpus to extract monolingual data	118
<i>Andrea Dömötör</i>	
Improving Surface-syntactic Universal Dependencies (SUD): MWEs and deep syntactic features	126
<i>Kim Gerdes, Bruno Guillaume, Sylvain Kahane and Guy Perrier</i>	
Artificially Evolved Chunks for Morphosyntactic Analysis	133
<i>Mark Anderson, David Vilares and Carlos Gómez-Rodríguez</i>	
Challenges of language change and variation: towards an extended treebank of Medieval French	144
<i>Mathilde Regnault, Sophie Prévost and Eric Villemonte de la Clergerie</i>	

Invited Talk

Thursday 29th August 2019

Quantitative Computational Syntax: dependencies, intervention effects and word embeddings

Paola Merlo

University of Geneva

Abstract

In the computational study of intelligent behaviour, the domain of language is distinguished by the complexity of the representations and the vast amounts of quantitative text-driven data. In this talk, I will let these two aspects of the study of language inform each other and will discuss current work investigating whether the notion of similarity in the intervention theory of locality is related to current notions of similarity in word embedding space.

Despite their practical success and impressive performances, neural-network-based and distributed semantics techniques have often been criticized as they remain fundamentally opaque and difficult to interpret. Several recent pieces of work have investigated the linguistic abilities of these representations, and shown that they can capture long agreement and thus hierarchical notions. In this vein, we study another core, defining and more challenging property of language: the ability to construe long-distance dependencies. We present results that show that word embeddings and the similarity spaces they define do not correlate with experimental results on intervention similarity in long-distance dependencies. These results show that the linguistic encoding in distributed representations does not appear to be human-like, and it also brings evidence to the debate on narrow or broad definitions of similarity in syntax and sentence processing.

Short bio

Paola Merlo is associate professor in the Linguistics department of the University of Geneva. She is the head of the interdisciplinary research group Computational Learning and Computational Linguistics (CLCL). The group is concerned with interdisciplinary research combining linguistic modelling with machine learning techniques. Prof. Merlo has been editor of Computational Linguistics, published by MIT Press and a member of the executive committee of the ACL. Prof. Merlo holds a doctorate in Computational Linguistics from the University of Maryland, and has been associate research fellow at the University of Pennsylvania, and visiting scholar at Rutgers, Edinburgh, Stanford and Uppsala.

Are formal restrictions on crossing dependencies epiphenomenal?

Himanshu Yadav
IIT Kanpur
Department of Humanities
and Social Sciences
yadavhimanshu059@gmail.com

Samar Husain*
IIT Delhi
Department of Humanities
and Social Sciences
samar@hss.iitd.ac.in

Richard Futrell*
University of California, Irvine
Department of
Language Science
rfutrell@uci.edu

Abstract

Characterizing the distribution of crossing dependencies in natural language dependency trees is a crucial task for building parsers and understanding the formal properties of human language. A number of formal restrictions on crossing dependencies have been proposed, including bounds on gap degree, edge degree, and end-point crossings. Here we ask whether the empirical distribution of crossing dependencies in dependency treebanks offers evidence for these formal restrictions as true, independent constraints on dependency trees, or whether the distribution can be explained using other, more generic constraints affecting dependency trees. Specifically, we explore the null hypothesis that crossing dependencies are formally unrestricted, but occur at a low rate. We implement the null hypothesis using random trees where crossing dependencies occur at the same rate as in natural language trees, but without any formal restrictions. We find that this baseline generally does not reproduce the same distribution of gap degree, edge degree, end-point-crossing, and heads' depth difference as real trees, suggesting that these formal constraints are a consequence of factors beyond the rate of crossing dependencies alone.

1 Introduction

In dependency grammar formalisms, the syntactic structure of a sentence is encoded in the form of head-dependent relations. For the most part, the dependents of a given head form a contiguous substring of the sentence, i.e., all the nodes occurring between the head and its dependent are (transitively) dominated by the head. Such dependencies have been termed **projective**. In addition to projective dependencies, we also find instances where the dependents of a head are discontinuous. This happens when a node in the span of a head and its dependents is not (directly or indirectly) dominated by the head. Such dependencies are known as **crossing** or **non-projective dependencies**. Formally, a dependency $X_h \rightarrow X_d$ is deemed crossing if and only if there is at least one node X_i between X_h and X_d that X_h does not dominate. In Figure 1 the dependency arc from the node X_h to its dependent X_d is crossing because X_i is headed by a node (X_j) which is outside the span of $X_h \rightarrow X_d$. Note that all other arcs in the dependency tree shown in Figure 1 are projective. For example, the arc $X_j \rightarrow X_i$ is a projective arc as X_h is dominated by X_j .

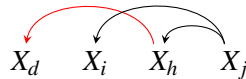


Figure 1: The dependency arc $X_h \rightarrow X_d$ is a crossing dependency. All other arcs are non-crossing.

The most basic cross-linguistic generalization about crossing dependencies is that they are rare (see e.g. Straka et al., 2015). The rarity of crossing dependencies poses several interesting questions that are relevant from formal, computational, and cognitive perspectives. Most fundamentally, why are these

* Equal contribution

constructions rare? When and why are these constructions difficult for computational parsers and humans? Are there general constraints on the space of variation in natural languages that can explain this rarity?

Investigating the constraints which cause the rarity of crossing dependencies could help us in discovering the underlying principles that have shaped human language. Not surprisingly, there have been previous attempts to investigate the cause of this rarity formally as well as from a processing perspective (e.g., Shieber, 1985; Bach et al., 1986; Vogel et al., 1996; Ferrer-i-Cancho, 2006; Levy et al., 2012; Kuhlmann, 2013; Husain and Vasishth, 2015; Ferrer-i-Cancho and Gómez-Rodríguez, 2016; Yadav et al., 2017, under review). In addition, a number of formal restrictions on crossing dependencies have also been proposed. Kuhlmann (2013) proposes that dependency trees have limited *gap degree* and are usually *well-nested* (see Figure 2b). Pitler et al. (2013) propose that crossing dependency configurations have a property called *1-end-point-crossing*. Other formal restrictions such as *edge degree*, *multiplicity* and *heads' depth difference* have also been proposed (Yli-Jyrä, 2003; Kuhlmann and Nivre, 2006; Nivre, 2007; Yadav et al., 2017). In this paper, we call these formal constraints on crossing dependencies **crossing constraints**.



Figure 2: The **projection chain** of a node X is the set of all the nodes dominated by X which lie in a single path from X to a terminal node. For example, in the dependency tree (a), $\{X_j, X_h, X_d, X_g\}$ and $\{X_j, X_i, X_k\}$ are two projection chains from the node X_j . A projection chain is continuous if it forms a continuous substring of the sentence. For example, the projection chain of X_h , i.e., $\{X_h, X_d, X_g\}$ is a continuous substring of the sentence $\{X_k, X_i, X_g, X_d, X_h, X_j\}$. The dependency tree (b) shows a dependency schema to illustrate gap degree. The gap degree of a node is the largest number of discontinuities in any projection chain. In (b), the projection chain for X_h is $\{X_h, X_d, X_g\}$, which contains 2 discontinuities or gaps, so the gap degree of node X_j is 2.

Crossing constraints are important in two domains: in the development of computational parsers, and for theoretical formal syntax, because these restrictions correspond to the formal language class of natural language. Crossing dependencies indicate deviations from context-free grammar (Marcus, 1965; Shieber, 1985). More specifically, the hierarchy of mildly context-sensitive languages is defined by restrictions on gap degree. Gap degree corresponds to the number of components in a Multiple Context-Free Grammar (Seki et al., 1991) and to the number of distinct selector features in Minimalist Grammars (Michaelis, 1998). It corresponds to the ‘limited amount of cross-serial dependencies’ allowed in TAG derivations (Joshi, 1985), (also see Bodirsky et al., 2005). In the computational linguistics literature it is common to provide statistics showing that there are only a small number of dependency trees violating any given crossing constraint. For example, Kuhlmann (2013) shows that as gap degree increases, there are fewer and fewer trees per language with that gap degree.

These proposals across the theoretical syntax and parsing literature raise the possibility that crossing constraints might constitute independent, causal constraints on natural language syntax. However, it is also possible that the observed distribution of crossing dependencies may be epiphenomenal, i.e., a consequence of other constraints affecting dependency trees which have nothing to do with crossing dependencies themselves, such as a general pressure to minimize dependency length (e.g., as investigated in Ferrer-i-Cancho and Gómez-Rodríguez, 2016; Gómez-Rodríguez and Ferrer-i-Cancho, 2017). In this paper, we investigate the status of crossing constraints using dependency corpora, asking whether the empirical distribution of crossing dependencies gives evidence for crossing constraints, or whether the data is best explained by an extremely simple null hypothesis: that crossing dependencies are formally unrestricted but simply rare.

As an example of how crossing constraints might be epiphenomenal, consider gap degree. Gap degree refers to the number of discontinuities in the projection chain headed by a node (see Figure 2). So, for example, if the longest projection chain in a sentence is of length 6, then gap degree cannot exceed 5. Now suppose that linguistic dependency trees typically have short projection chains and that crossing dependencies are rare but randomly distributed across dependency trees. Then it is unlikely that we will observe a projection with many discontinuities, simply due to the fact that projection chains are usually short; so we will measure low gap degree. From this measurement, we might falsely conclude that there exists a bound on gap degree. These considerations suggest that gap degree might not have a causal role as a restriction on crossing dependencies, but rather emerges as a result of the rarity of crossing dependencies plus low tree depth.

In this work, we evaluate a number of crossing constraints to determine if dependency corpora give evidence for them as true independent constraints. Our **null hypothesis** is that crossing dependencies are formally unrestricted, but occur at a certain low rate per dependency arc. The alternative to the null hypothesis is the **true constraint hypothesis** (TCH), which is that there is a real dispreference for crossing dependencies violating that specific constraint, arising from grammar or cognitive pressures.

We compare the TCH against the null hypothesis by comparing natural language dependency trees with randomly generated baseline trees. The baseline trees simulate the null hypothesis: they consist of randomly generated trees where crossing dependencies have been inserted randomly at the same overall rate per dependency as in the real trees, but with no formal restrictions (more on this in Section 3.2). If the distribution of gap degree, edge degree, etc., in random baseline trees is indistinguishable from real language trees, then we cannot reject the null hypothesis: in that case dependency corpora would not show evidence for the TCH. On the other hand, if a formal measure like gap degree is minimized in observed data over the random baseline, then this is evidence against the null hypothesis and for the TCH.

Our paper is organized as follows. In Section 2, we review the crossing constraints that we will test. In Section 3, we discuss the natural language dataset and the random baselines. We present the results in Section 4. Section 5 concludes.

2 Measures

In order to test the TCH, we compare the distributions of violations of crossing constraints in random baseline trees vs. real language trees. Below we discuss the crossing constraints used in our investigation. In addition, we also discuss the properties of the dependency tree that are used in our comparison of real vs. random trees. In particular, we will be testing whether the correlation between these dependency tree properties and crossing constraint violations is the same in real vs. random trees.

2.1 Crossing Constraints

Gap degree: The **gap degree** of a node X is the number of discontinuities in the projection of node X . For example, in Figure 2, the projection chain of node X_h contains two discontinuities; these discontinuities are present in $X_h \rightarrow X_d$ and in $X_d \rightarrow X_g$. Therefore, the gap degree of node X_h is 2. On the other hand, the gap degree of node X_d is 1. The gap degree of a dependency tree is the maximum among the gap degrees of its nodes (Kuhlmann and Nivre, 2006). In Figure 2, the gap degree of the tree is 2 as the highest gap degree (associated with X_h) is 2. Since gap degree is number of discontinuities in a projection chain, it is upper bounded by the length of projections chains.

Edge degree: Let e be the span of dependency arc $X_h \rightarrow X_d$. The span e consists of nodes between a head X_h and its dependent X_d , which are X_i , X_a , and X_b in Figure 3. The **edge degree** of a dependency arc $X_h \rightarrow X_d$ is the number of nodes in the span e which are neither dominated by some node in the span e nor dominated by the head X_h . For example, arc $X_h \rightarrow X_d$ in Figure 3(a) and 3(b) has an edge degree of 2 because node X_i and X_b are not dominated by any node in the span e . In addition, they are also not dominated by head X_h . The edge degree of a dependency tree is the highest edge degree among the arcs of the tree.

There are cognitive reasons to suspect edge degree might be limited in natural language. From an on-line processing perspective, higher edge degree in a subtree results in the need to maintain an unresolved crossing dependency across a longer span of words, which may result in online processing difficulty due to higher working memory load (Gibson, 1998).

End-point crossing: The number of **end-point crossings** is the number of heads which dominate the gap of an arc. Given an arc $X_h \rightarrow X_d$ with a span e containing X_i, X_a and X_b as in Figure 3, the end-point crossing of arc $X_h \rightarrow X_d$ is defined as the number of heads modified by the nodes in e that are not part of the projection chain of X_h . For example, in Figure 3(a) and 3(b), X_i and X_b are not part of the projection chain of X_h , in other words they are not dominated by either X_h or any node in the span e . In 3(a), the number of heads modified by X_i and X_b is 1 (corresponding to X_j), therefore, the end-point crossing is 1. In 3(b), the number of heads modified by X_i and X_b are 2 (corresponding to X_j and X_r respectively), therefore, the end-point crossing is 2.

It has been argued that natural language dependency trees tend to have not more than one end-point crossing, which is called the 1-end-point crossing constraint (Pitler et al., 2013). Pitler et al. (2013) argue that this constraint is related to the Phase Impenetrability Condition from Minimalist syntax (Chomsky, 2007). From a processing based perspective, higher end-point crossings in a subtree should lead to multiple heads/dependents being maintained/stored at the same time in the parse stack. This should lead to increased storage cost (Gibson, 1998). In addition, a longer span of the crossing dependency could lead to similarity-based interference (Lewis and Vasishth, 2005) at the head.

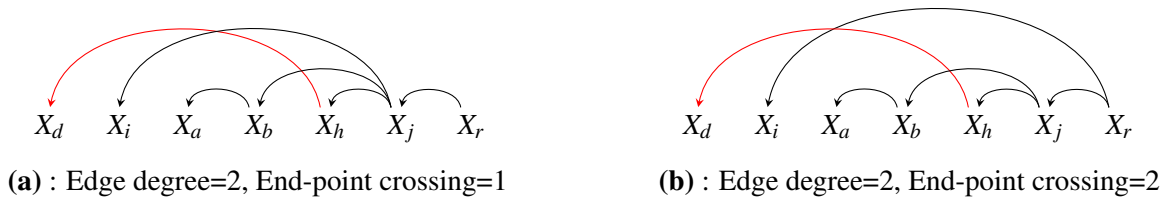


Figure 3: Dependency schemas showing edge degree and end-point crossing. In both the dependency trees, $X_h \rightarrow X_d$ is a crossing dependency. The span of crossing dependency e consists of X_i, X_a and X_b . X_i and X_b are dominated neither by head X_h nor by any node in span e . In (a) and (b), different sets of nodes are modified by X_i and X_b .

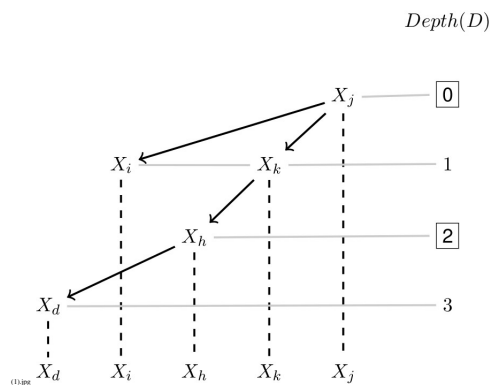


Figure 4: A schematic diagram for heads' depth difference (HDD).

Heads' depth difference (HDD): For a crossing dependency $X_h \rightarrow X_d$, suppose that X_i is the node which creates discontinuity, i.e. X_i is not directly or indirectly dominated by X_h (see Figure 4). For this configuration, we call X_i the **intervener**, X_j the head of the intervener, and X_h the head of the crossing dependency. The **heads' depth difference (HDD)** is defined as the difference between the depth of head of the crossing dependency X_h and depth of head of the intervener X_j . This is schematically shown in

Figure 4. Depth of a node is computed as the hierarchical position of that node in a projection chain. The depth of X_h is 2 while the depth of X_j is 0, making the HDD for this configuration equal to 2. Thus, HDD for a crossing dependency $X_h \rightarrow X_d$ is:

$$\text{HDD}(X_h, X_d) = \text{depth}(X_h) - \text{depth}(X_j), \quad (1)$$

where $\text{depth}(X_h)$ is the hierarchical position of the head of the non-projective dependency (X_h) and $\text{depth}(X_j)$ is the hierarchical position of the head of the intervening element (X_j). The HDD of a dependency tree is the maximum HDD among the HDDs of the arcs in the tree.

In terms of formal syntax, HDD can correspond to the hierarchical depth between a filler and a gap in a long distance dependency (e.g., wh movement). Based on the theoretical syntax literature, HDD should be unbounded, at least for leftward wh-dependencies (Sag et al., 1999). However, increasing HDD seems to correlate with increased online processing difficulty for humans (Phillips et al., 2005). More generally, HDD has been proposed (see Yadav et al., 2017) to formalize the experimental findings that increased embedding depth leads to processing difficulty (e.g., Yngve, 1960; Gibson and Thomas, 1999). Therefore, it is possible that HDD is restricted in dependency trees due to cognitive constraints.

2.2 Dependency tree properties

We study violations of crossing constraint as a function of the following properties of dependency trees.

Sentence length: Sentence length is measured as the total number of nodes in a dependency tree.

Arity: The arity of a node is the total number of dependents of that node. We quantify arity as a global property of a tree by taking the maximum arity per node in the tree.

Tree depth: Tree depth is the number of heads in the longest projection chain in a dependency tree (see Figure 2). Tree depth represents the maximum number of levels of embedding occurring in a tree.

3 Data and methodology

3.1 Natural languages dataset

We use the Universal Dependencies (UD v2.3) treebanks of 14 languages as a dataset (Nivre et al., 2018). The languages were selected for typological diversity: the dataset contains 8 head-initial languages and 6 head-final languages. We do not include dependencies marking punctuation (labeled as ‘punct’ in UD scheme) and the abstract root of the tree (labeled as ‘root’ in UD scheme) in our analysis.

As we discuss below, the process of sampling random baseline trees makes it prohibitively difficult to study all languages in the UD dataset. Therefore we study treebanks of 14 languages: German, English, Hindi, French, Arabic, Russian, Czech, Italian, Spanish, Afrikaans, Japanese, Korean, Bulgarian and Slovak. We present results aggregating over dependency trees from all these languages.

3.2 Random baseline

Our null hypothesis is that the only restriction on crossing dependencies is that they are rare, i.e. that they occur at some certain low rate per dependency in a sentence. We instantiate the null hypothesis by sampling random trees which are constrained to have the same distribution over sentence length and number of crossings per dependency as a corpus of some natural language.

We control for sentence length and crossing rate in the random trees in the following way. For each real dependency tree t of length n in a corpus, we sample random trees t' from a uniform distribution over n^{n-1} directed labeled tree structures with n nodes using Prüfer codes (Prüfer, 1918). We control for the crossing rate by rejection sampling: we reject random samples t' which do not have the same number of crossings as the original tree t . For long sentences (over length 12), the rejection sampling process is prohibitively slow, because the vast majority of random trees for $n \geq 12$ have a very large number of crossings. So in the present work we only consider sentences of length less than 12.

Since we are only controlling the number of crossings and the sentence length, the distribution of arity and depth in random baseline trees is quite different from real language trees. In particular, we find that

the growth of tree depth with respect to sentence length is faster for random baseline trees. In addition, the growth of arity with sentence length in the random tree is slower. In sum, random baseline trees are typically deeper than real trees.

3.3 Testing the Null and True Constraint Hypotheses

We compare the rate at which crossing constraints are violated in real trees as compared with random baseline trees, as a function of sentence length, arity, and tree depth. We evaluate the difference between real and random trees statistically using mixed-effects Poisson regression (Gelman and Hill, 2007; Baayen et al., 2008). We fit the regression to predict the rate of constraint violations as a function of dependency tree features (length, depth, and arity) and a dummy-coded variable encoding whether a given tree is real or random. We also include by-language random intercepts. For example, we predict the gap degree g_i of the i th sentence s_i in the j th language as:

$$\log E[g_i] = \beta_0 + \beta_l |s_i| + \beta_r r_i + \beta_{lr} r_i |s_i| + \gamma_j + \epsilon, \quad (2)$$

where $|s_i|$ is the length of sentence s_i , r_i is an indicator variable with value 1 for a real tree and 0 for a baseline tree, and γ_j , subject to L_2 regularization, is a random intercept for the j th language. The fitted value of the **interaction coefficient** β_{lr} gives the extent to which the growth rate of gap degree as a function of sentence length differs between the real and the random trees. If β_{lr} is significantly negative, then this would mean that gap degree grows more slowly with sentence length in real trees as compared with random trees, i.e. gap degree would be minimized in real trees.

4 Results

We compared the regression pattern of each measure with length, arity and depth between observed and random baseline trees. Below we report the results for each crossing constraint. A summary of all regression results is found in Table 1.

4.1 Gap degree

We find that the distribution of gap degree as a function of sentence length and arity is not significantly different between real and random trees (see Figure 5). In particular, the interaction between length/arity and tree type was not significant in the respective models (see Table 1). However, growth rate of gap degree with tree depth is significantly different between real and random trees ($p < .001$). In other words, we found no evidence for the TCH for gap degree as a function of length and arity: the distribution of gap degree in natural language trees can be fully explained without formal restrictions on crossing dependencies or tree structures. However, the results with respect to depth provide support for the TCH.

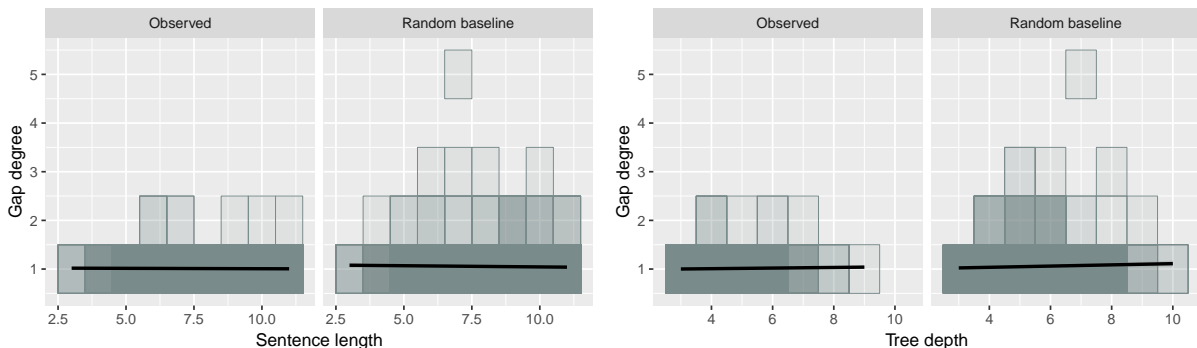


Figure 5: Gap degree as a function of sentence length and tree depth in real and random trees. In this and all other figures, for visual clarity, we only display results for trees with at least one crossing dependency. All statistical tests are performed using all trees.

4.2 Edge degree

As shown in Figure 6, edge degree grows faster in random trees in comparison to real trees as a function of sentence length, arity and depth. The mixed-effects Poisson regression models show that the three interaction coefficients (for length, arity, and depth) are significant in the respective models (see Table 1). This provides evidence for the TCH for edge degree.

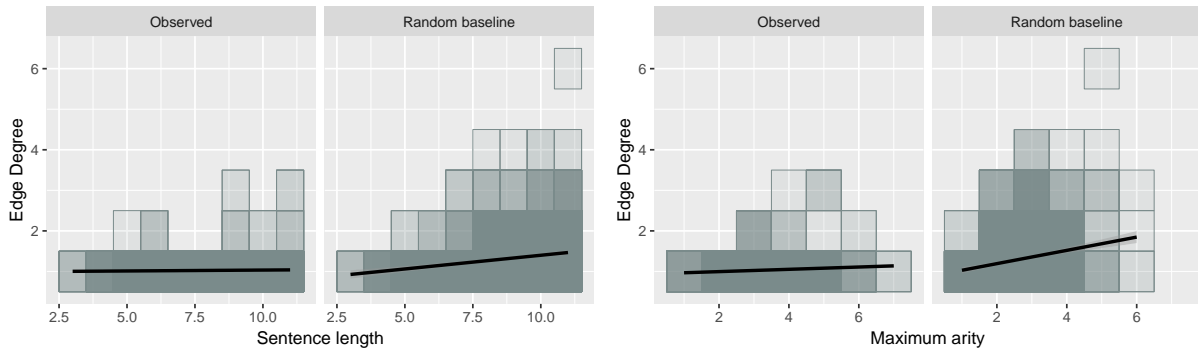


Figure 6: Edge degree as a function of sentence length and tree maximum arity for real and random trees.

4.3 End-point crossings

As shown in Figure 7, we find that end-point crossings grow at a slower rate in real trees as a function of tree depth as compared with random baselines. The results support the TCH for end-point crossings. Similar to gap degree, end-point crossing as a function of maximum arity and sentence length does not differ significantly between real and random trees (see Table 1).

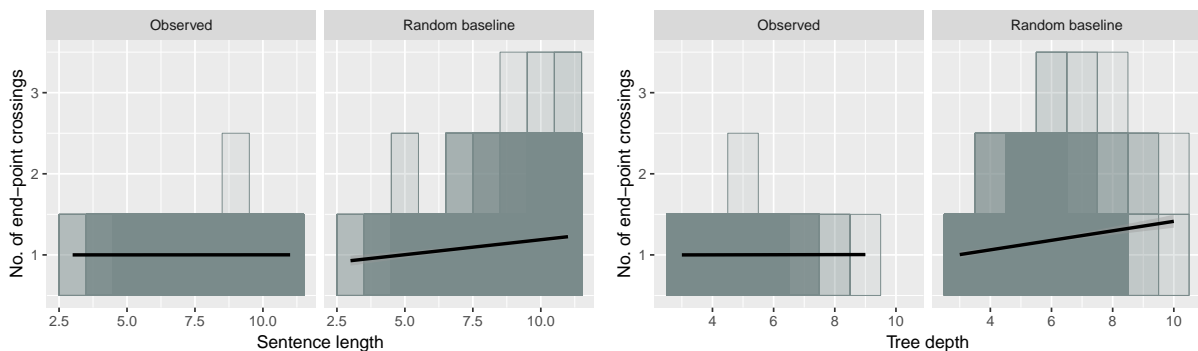


Figure 7: End-point crossings as a function of sentence length and tree depth in real and random trees.

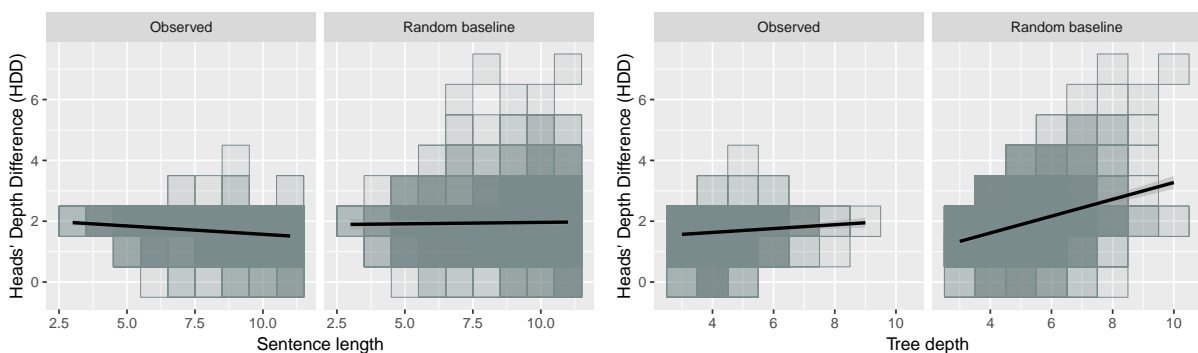


Figure 8: HDD as a function of sentence length and tree depth in real and random trees.

4.4 Heads' Depth Difference (HDD)

The results show that HDD decreases with sentence length in real trees, and the rate of decrease is less than in random trees (Figure 8). HDD is also much higher in random trees compared to real trees as a function of tree depth. These results support the TCH for HDD. HDD does not differ between real and random tree with respect to maximum arity (see Table 1).

Dependent variable	Independent variable	β Estimate	Std. Error	p value	
Gap degree	Sentence length	0.75	0.04	< 2e-16	*
	Observed	-0.07	0.05	0.174	n.s.
	Sentence length \times Observed	-0.03	0.05	0.563	n.s.
	Arity	0.25	0.03	2.88e-14	*
	Observed	-0.18	0.04	0.00013	*
	Arity \times Observed	-0.06	0.04	0.1570	n.s.
	Depth	0.52	0.02	< 2e-16	*
	Observed	0.24	0.05	1.23e-05	*
	Depth \times Observed	0.29	0.04	2.43e-10	*
Edge degree	Sentence length	0.37	0.03	< 2e-16	*
	Observed	-0.20	0.04	1.41e-06	*
	Sentence length \times Observed	-0.11	0.04	0.0153	*
	Arity	0.09	0.02	0.0015	*
	Observed	-0.24	0.04	3.65e-09	*
	Arity \times Observed	-0.13	0.04	0.0009	*
	Depth	0.32	0.02	< 2e-16	*
	Observed	0.04	0.04	0.33	n.s.
	Depth \times Observed	0.21	0.04	1.02e-06	*
End-point crossing	Sentence length	0.32	0.03	< 2e-16	*
	Observed	-0.10	0.04	0.0173	*
	Sentence length \times Observed	-0.07	0.04	0.1013	n.s.
	Arity	-0.001	0.03	0.9900	n.s.
	Observed	-0.10	0.04	0.0141	*
	Arity \times Observed	-0.07	0.04	0.1098	n.s.
	Depth	0.34	0.02	< 2e-16	*
	Observed	0.16	0.04	0.0002	*
	Depth \times Observed	0.20	0.04	3.92e-06	*
HDD	Sentence length	0.27	0.02	< 2e-16	*
	Observed	-0.14	0.03	8.78e-06	*
	Sentence length \times Observed	-0.08	0.03	0.0152	*
	Arity	-0.11	0.02	7.36e-06	*
	Observed	-0.10	0.03	0.0025	*
	Arity \times Observed	-0.02	0.03	0.4695	n.s.
	Depth	0.44	0.02	< 2e-16	*
	Observed	0.21	0.03	1.19e-08	*
	Depth \times Observed	0.13	0.03	8.78e-06	*

Table 1: Mixed-effect Poisson regression results for all the crossing constraints and dependency tree measures for 14 languages. “Observed” is an indicator variable with value 1 for observed trees and 0 for random trees, the same as r_i in Equation 2. A significant interaction between an independent variable and Observed rejects the null hypothesis.

5 Conclusion

We found that the distribution of gap degree, edge degree, end-point crossing and HDD cannot be explained solely in terms of sentence length and the rate of crossings. These constraints are violated at a different rate as a function of various tree properties than would be expected in random trees, suggesting that they may constitute real formal restrictions on trees.

The results show that the behavior of these crossing constraints differ depending dependency tree properties. Gap-degree and end-point crossings in real vs. random trees are only different as a function of tree depth (which itself has a very different distribution between real and random trees). HDD in real vs random trees is indistinguishable as a function of arity, but is different for tree depth and sentence length. Edge degree, on the other hand, emerges as the crossing constraint that is most distinct between real and random trees: its distribution is significantly different as a function of all three tree properties.

Our results do not rule out the possibility that the correlations reported here might themselves be epiphenomenal, resulting from other graph-theoretic properties of real dependency trees which were not controlled for here. For example, a great deal of work has shown that syntactic dependency trees are subject to dependency length minimization: a pressure for the linear distance between syntactic heads and dependents to be short (Hawkins, 1994; Gibson, 1998; Liu, 2008; Futrell et al., 2015) (for recent reviews, see Liu et al., 2017; Temperley and Gildea, 2018), and this pressure has been argued to underly the scarcity of crossing dependencies in general (Ferrer-i-Cancho, 2006; Ferrer-i-Cancho and Gómez-Rodríguez, 2016; Gómez-Rodríguez and Ferrer-i-Cancho, 2017). It is also possible that the differences between real trees and random trees in our results are driven by differences in the depth and arity of these trees, or by UD annotation decisions such as the use of content-head dependencies.

Our work provides a strong framework for evaluating any such theory that aims to predict the particular distribution of crossing dependencies in natural language. A syntactic theory can be tested in our framework by creating random baselines that control for the stipulations of the theory and then statistically comparing the distribution of crossing constraint violations with real trees. To that end, we make the code for our analysis freely available at http://github.com/yadavhimanshu059/measures_of_nonProjectivity.

Acknowledgments

The authors thank Tim O’Donnell for helpful discussion and the anonymous reviewers for helpful comments on the paper.

References

- R. Harald Baayen, D.J. Davidson, and Douglas M. Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59(4):390–412.
- Emmon Bach, Colin Brown, and William D. Marslen-Wilson. 1986. Cross and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes*, 1(4):249–262.
- Manuel Bodirsky, Marco Kuhlmann, and Mathias Möhl. 2005. Well-nested drawings as models of syntactic structure. In *In Tenth Conference on Formal Grammar and Ninth Meeting on Mathematics of Language*, pages 195–203.
- Noam Chomsky. 2007. Approaching UG from below. In *Interfaces + recursion = language?: Chomsky’s Minimalism and the view from syntax-semantics*, pages 1–29. Mouton de Gruyter.
- Ramon Ferrer-i-Cancho. 2006. Why do syntactic links not cross? *Europhysics Letters*, 76(6):1228.
- Ramon Ferrer-i-Cancho and Carlos Gómez-Rodríguez. 2016. Crossings as a side effect of dependency lengths. *Complexity*, 21(S2):320–328.
- Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. <https://doi.org/10.1073/pnas.1502134112> Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

- Andrew Gelman and Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, Cambridge, UK.
- Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.
- Edward Gibson and James Thomas. 1999. Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical. *Language and Cognitive Processes*, 14(3):225–248.
- Carlos Gómez-Rodríguez and Ramon Ferrer-i-Cancho. 2017. Scarcity of crossing dependencies: A direct outcome of a specific constraint? *Physical Review E*, 96.
- John A. Hawkins. 1994. *A performance theory of order and constituency*. Cambridge University Press, Cambridge.
- Samar Husain and Shravan Vasishth. 2015. Non-projectivity and processing constraints: Insights from Hindi. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, Uppsala, Sweden.
- Aravind K. Joshi. 1985. Processing of sentences with intrasentential code switching. In D. R. Dowty, L. Karttunen, and A. M. Zwicky, editors, *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, pages 190–205. Cambridge University Press, Cambridge.
- Marco Kuhlmann. 2013. Mildly non-projective dependency grammar. *Computational Linguistics*, 39(2):355–387.
- Marco Kuhlmann and Joakim Nivre. 2006. Mildly non-projective dependency structures. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 507–514, Sydney, Australia. Association for Computational Linguistics.
- Roger P. Levy, Evelina Fedorenko, Mara Breen, and Edward Gibson. 2012. The processing of extraposed structures in English. *Cognition*, 122(1):12–36.
- Richard L. Lewis and Shravan Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29(3):375–419.
- Haitao Liu. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2):159–191.
- Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of Life Reviews*.
- Solomon Marcus. 1965. Sur la notion de projectivité. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 11(2):181–192.
- Jens Michaelis. 1998. Derivational Minimalism is mildly context-sensitive. In *Logical Aspects of Computational Linguistics*, volume 98, pages 179–198. Springer.
- Joakim Nivre. 2007. Incremental non-projective dependency parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, Rochester, New York. Association for Computational Linguistics.
- Joakim Nivre, Mitchell Abrams, et al. 2018. <http://hdl.handle.net/11234/1-2895> Universal dependencies 2.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Colin Phillips, Nina Kazanina, and Shani H. Abada. 2005. ERP effects of the processing of syntactic long-distance dependencies. *Cognitive Brain Research*, 22(3):407–428.
- Emily Pitler, Sampath Kannan, and Mitchell Marcus. 2013. Finding optimal 1-endpoint-crossing trees. *Transactions of the Association for Computational Linguistics*, 1:13–24.
- Heinz Prüfer. 1918. Neuer Beweis eines Satzes über Permutationen. *Archiv der Mathematischen Physik*, 27:742–744.

- Ivan A. Sag, Thomas Wasow, and Emily M. Bender. 1999. *Syntactic theory: A formal introduction*. Center for the Study of Language and Information, Stanford, CA.
- Hiroyuki Seki, Takashi Matsumara, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):191–229.
- Stuart M. Shieber. 1985. Evidence against the context-freeness of natural language. In *The Formal complexity of natural language*, pages 320–334. Springer.
- Milan Straka, Jan Hajič, Jana Straková, and Jan Hajič, Jr. 2015. Parsing universal dependency treebanks using neural networks and search-based oracle. In *14th International Workshop on Treebanks and Linguistic Theories (TLT 2015)*, pages 208–220, Warszawa, Poland. IPIPAN.
- David Temperley and Dan Gildea. 2018. Minimizing syntactic dependency lengths: Typological/cognitive universal? *Annual Review of Linguistics*, 4:1–15.
- Carl Vogel, Ulrike Hahn, and Holly Branigan. 1996. Cross-serial dependencies are not hard to process. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 157–162. Association for Computational Linguistics.
- Himanshu Yadav, Ashwini Vaidya, and Samar Husain. 2017. Understanding constraints on non-projectivity using novel measures. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, Pisa, Italy. Linköping University Electronic Press.
- Himanshu Yadav, Ashwini Vaidya, Vishakha Shukla, and Samar Husain. under review. Word order typology interacts with linguistic complexity: a cross-linguistic corpus study.
- Anssi Mikael Yli-Jyrä. 2003. Multiplanarity—a model for dependency structures in treebanks. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, pages 189–200, Vaxjö, Sweden. Vaxjö University Press.
- Victor H. Yngve. 1960. A model and an hypothesis for language structure. *Proceedings of the American Philosophical Society*, 104(5):444–466.

A Surface-Syntactic UD Treebank for Naija

Bernard Caron
IFRA, CNRS
bernard.caron
@cnrs.fr

Marine Courtin **Kim Gerdes**
LPP, Sorbonne Nouvelle & CNRS
kim@gerdes.fr,
marine.courtin
@sorbonne-nouvelle.fr

Sylvain Kahane
Modyco, Université
Paris Nanterre & CNRS
sylvain@kahane.fr

Abstract

This paper presents a syntactic treebank for spoken Naija, an English pidgincreole, which is rapidly spreading across Nigeria. The syntactic annotation is developed in the Surface-Syntactic Universal Dependency annotation scheme (SUD) (Gerdes et al., 2018) and automatically converted into UD. We present the workflow of the treebank development for this under-resourced language. A crucial step in the syntactic analysis of a spoken language consists in manually adding a markup onto the transcription, indicating the segmentation into major syntactic units and their internal structure. We show that this so-called “macrosyntactic” markup improves parsing results. We also study some iconic syntactic phenomena that clearly distinguish Naija from English.

1 Introduction

Naija is an English pidgincreole (Bakker, 2009) spoken by an estimated 100 million speakers in Nigeria and the Nigerian diaspora in Africa, the UK and the USA. Its origin lies in Nigerian Pidgin, a creole spoken in the Niger delta (Faraclas, 1989; Elugbe and Omamor, 1991). As the creole escaped its ecological niche and spread all over Nigeria since the national independence (1960), it has acquired new functions and is spoken as a second language by speakers whose first language belongs to the four genetic phyla represented by the 500 or so languages spoken in Nigeria. Although it has no official status or standard orthography, it has been adopted for private and informal communication by the educated youth and the Nigerian elite. The Wazobia radio and TV network founded in 2007, uses Naija as its only medium, and the BBC opened a “Pidgin” station in Lagos in 2017, “Pidgin” being the common name used by the locals to name what we call “Naija”.

In the process, Naija has developed new structures, a new vocabulary, and probably a new prosody, that differentiate it from Nigerian Pidgin. Despite the ever-growing importance of the language in Nigeria, little attention has been paid to Naija as such, and most of what can be read in the literature concerning the language is based on impressionistic intuitions influenced by previous descriptions of Nigerian Pidgin. This has driven us to start the NaijaSynCor project (NSC), a corpus-based survey of Naija, financed by the French research agency ANR (Caron, 2017). The size of the language, and its geographical span has induced a specific choice of variationist sociolinguistics (Tagliamonte, 2012) as a theoretical framework, and an extensive use of Natural Language Processing tools for our corpus annotation and interpretation.

As it stands now, the NSC corpus counts 321 audio files averaging 5 minutes each, and 319 speakers, which represents a total of 500,000 words collected in 11 locations (see Figure 1). The genres recorded cover life stories, speeches, radio programs, free conversations, cooking recipes, comments on current state of affairs, etc. The sampling of speakers aims at balancing age, sex, education, linguistic and geographic background. Our aim is to annotate each file as finely as possible and prepare queries that cross the linguistic annotation with demographic information collected from each of the 319 speakers. The audio files are annotated with time-aligned transcription and translation into English, morphological tagging, macrosyntactic segmentation, dependency syntax, and prosodic annotation.



Figure 1. Map of the 11 survey locations

This paper focuses on the syntactic annotation of the corpus and the constitution of a 150,000 words gold standard treebank. The current state of the treebank is accessible at SurfaceSyntacticUD.github.io and can be queried directly at http://match.grew.fr/?corpus=SUD_Naija-NSC@dev. The treebank is currently still undergoing manual and automatic validation. An automatically converted UD version of our treebank will extend the current UD Naija-NSC (available since UD 2.2 (Nivre et al., 2018)) with the upcoming release.

Once the 150k words gold section will be completed, the rest of the corpus will be parsed automatically, together with the 250k words of spoken (historic) Nigerian Pidgin data from Deuber (2005), resulting in a treebank of about 750k words.

Our workflow is explained in section 2, especially the choice of Surface-Syntactic UD, rather than UD. Section 3 presents some interesting constructions in Naija.

2 Treebank development

Section 2.1 concerns the corpus itself (metadata, transcription, translation, and glossing). The particularities of the morphosyntactic annotation, due to the fact that Naija is an English lexifier pidgincreole, are described in Section 2.2. Section 2.3 presents the theoretical choice of our segmentation into maximal syntactic units for this spoken corpus. SUD annotation is developed in Section 2.4. Evaluation is presented in Section 2.5.

2.1 Corpus

Metadata. The variationist analysis we have chosen implies collecting samples representing different types of speakers, and different types of functions. A questionnaire was administered and recorded to provide the relevant metadata about the speakers: time, place and conditions of recording; sex, age, education, professional activity, geographic origin, linguistic background and history. The information was entered into an IMDI¹ database produced using the metadata editor Arbil² (Withers, 2012).

¹ ISLE Meta Data Initiative (IMDI) is a metadata standard to describe multi-media and multi-modal language resources.

² <https://tla.mpi.nl/tools/tla-tools/arbil/>

Transcription. Naija is commonly written, in particular on the internet, in forums, but also for example on the BBC website. Although an official orthography or normalization has not taken place, the speakers of Naija have strong opinions on how most words have to be spelled, and we decided to follow these evolving conventions. Mostly, the speakers prefer etymological orthography (i.e. inspired by the Standard English) modified for some emblematic Naija words for which specific spellings have developed, e.g. *wetin* ‘what’, *moda* ‘mother’, *fada* ‘father’, *dem* ‘they/them/plural marker’. We have used a specific orthography to disambiguate certain function words, e.g. *de* (a variant of *dem*) vs. *dey* (the imperfective auxiliary); *come* ‘to come’ and *con* (the consecutive auxiliary), *say* ‘to say’, and *sey* (the reported speech complementizer). As this emerging orthography is not stabilized, in order to avoid promoting an artificially authoritative norm, we have maintained all the variants in the transcriptions. An example is the word ‘thing’, which can be written *ting*, *tin*, *thing* by the annotators. These variants are associated to a common lemma *ting*, which could be changed later following statistical tendencies that will emerge.

Translation. The translation of all the sentences into English has been done by a team of native speakers of Naija, once the macrosyntactic analysis had been stabilized. It aims at remaining as faithful as possible to the structure and style of the original oral data, keeping the hesitations, repetitions, and general disfluencies. However, the translators have had to strike a balance between a tendency common in Nigerian academics to use erudite and abstruse vocabulary, and on the other hand the risk of using Nigerian English expressions and grammar that would not be understood by non-Nigerians (e.g. a general tendency to use *would* instead of *will* as a future auxiliary).

2.2 Morphosyntactic analysis

Glossing and POS tagging. To start the annotation process, a first sample text was tagged with a model trained on English. Insofar as most of the lexicon of Naija is borrowed from English, and its meaning is transparent, the glossing was kept to a minimum. Function words do not have glosses beyond their morphological features, and only Naija lexical innovations were glossed (e.g. *pikin* ‘child’, *patapata* ‘full’). The POS annotation was manually corrected and a first dictionary of the function words and most common lexical items of Naija was created, containing the form, some orthographic variants, the POS tag, and an English gloss if necessary. This dictionary was then used on a dozen text samples inside the Elan-Corpa tool (Chanard, 2014), an extended version of the Elan tool³ (Sloetjes and Wittenburg, 2008), which proposes the dictionary’s POS for each token for validation by the annotator. Through this semi-automatic process, the dictionary was enriched and later on used by the automatic tagger that was developed for the project⁴. The POS tags follow the UD conventions (Nivre et al., 2018) with the caveat that some changes were made to accommodate the specificities of the Naija system. For example, Naija has three copulas, among which two are tagged as VERB (*be* and *dey* ‘be’) and one is tagged as PART (*na* ‘it is’)⁵. Regularly, the POS tagger is trained again on the corrected tags and thus improved in a bootstrapping loop.

Annotation guidelines. The annotation process for the samples was organized collectively, where each file was assigned to one of the three annotators. They were allowed to discuss the difficult cases among each other. At the end of this process, the annotation was consolidated through the use of a dictionary that was controlled independently and applied to the corpus. The final adjudication was done by an expert adjudicator on every single file. In this process some amendments had to be discussed more widely in the SUD community. The annotators are asked to verify their annotations by means of an annotation guide and to report directly into the guide any decision that is not directly derived from it. We thus have an annotation guide that undergoes constant refinement. The same process was used for the dependency annotation, see Section 2.4.

To assess the quality of the annotation we verified the inter-annotator agreement on three samples composed altogether of 121 sentences. The pre-parsed sample was annotated independently by our three annotators without communication among them and then validated by the expert to obtain the gold annotation. This allows us to compare the inter-annotator agreement based on the pre-parsed structure and measure the difference on the tags and relations that have to be changed to obtain the gold annotation.

³ <https://tla.mpi.nl/tools/tla-tools/elan/>

⁴ The POS tags were provided by a model of the Mate parser (Bohnet, 2010), other morpho-syntactic features were added by means of a Wapiti-based CRF tagger (Tellier et al., 2010).

⁵ The copulas are converted in the POS AUX in UD according to the UD guidelines.

	Percentage of agreement			Percentage of agreement when the annotation differs from the pre-parsed annotation		
	A/B ⁶	A/C	B/C	A/B	A/C	B/C
UPOS	95	94	95	46	41	37
UAS	93	91	91	68	60	58
LAS	89	86	87	60	51	50

Table 1. Inter-annotator agreement scores

The agreement scores are then improved by the final adjudication, and our semi-automatic query of the corpus to look for inconsistencies using the *grew* tool.

2.3 Macrosyntactic segmentation

Our segmentation is based on a long tradition of the study of syntax of spoken production in Romance languages (Blanche-Benveniste et al., 1990; Cresti, 2000; Degand and Simon, 2009). Our maximal syntactic units are illocutionary units, that is, assertions, questions, and demands. We use the markup developed in the Rhapsodie project of annotation of spoken French (Deulofeu et al., 2010; Pietrandrea and Kahane, 2019), which is a kind of formalized punctuation. The delimiter for illocutionary units is //. Consider this extract from a sample illustrating the markup:

- (1) den you go dey wrap dat food { small lr small } // cut cocoyam //= cut dat uh & // take { cocoyam lc and yam } wey you don grind //=
'then you will wrap that food in small pieces, cut the cocoyam, cut that er... take the cocoyam and yam which you have ground.' [DEU_A05]

We also mark lists: the notation { X | Y } indicates that the phrase Y occupies the same syntactic position as X and piles up on X (Gerdes and Kahane, 2009). Four types of lists are considered: “lc” marks coordination (*cocoyam and yam*), “lr” marks (syntactic) reduplication (*small small* ‘very small’), “la” marks appositions (*John my friend*), and “ll” marks disfluencies and reformulation:

- (2) { some ll some } people dey ask [e good make man { get ll go } test im children ?//] //
'some, some people were asking: "Is it good for a man to get... go and test his children ?"'
 [ABJ_GWA_09_Journalism_48]

An illocutionary unit is organized around a nucleus that bears the illocutionary force and some optional and non-autonomous components we call ad-nuclei. The nucleus is separated from pre- and post-nuclei by the delimiters “<” and “>”:

- (3) and many of dem wey vote dat time < na because of internet //
'and many of those who voted at the time, it was because of the internet' [ABJ_GWA_09_Journalism_27]

Inserting the macrosyntactic annotation into the text is part of the segmentation of the transcription and constitutes a first coarse-grained syntactic analysis. The macrosyntactic annotation can be studied as such to quantify phenomena that are more typical for spoken language such as left and right dislocations and disfluencies. It is also geared for the direct study of the prosody-syntax interface (Liu et al., 2019). The macrosyntactic annotation improves parsing results (see Section 2.5) and it can easily be simplified into a standard punctuation.

2.4 SUD

Two different strands of thought, one rather practical, the other more theoretical, have led us to annotating the corpus not in the standard UD dependency annotation scheme but rather in the Surface-Syntactic UD scheme (SUD) (Gerdes et al., 2018).

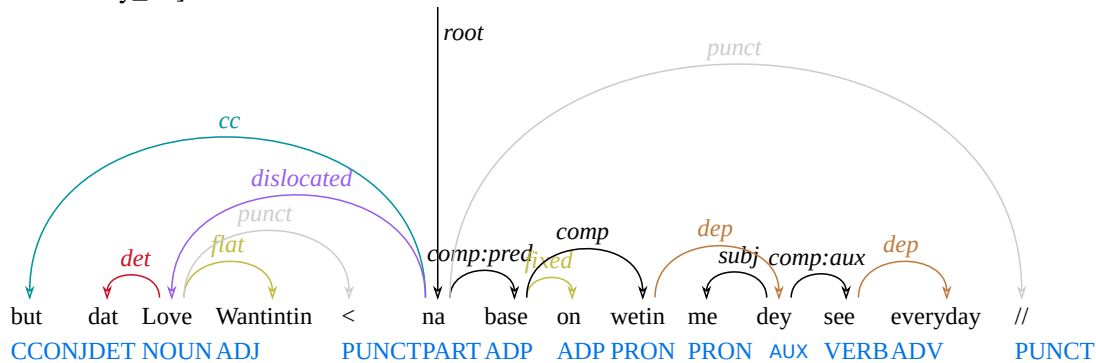
Firstly, the Nigerian annotators have been trained in a standard syntactic X-bar sentence structure, where, for example, a PP is headed by a preposition (Osborne and Gerdes, 2019). In this context, SUD is much easier to acquire than UD dependencies (Gerdes et al., 2019).

⁶ We look at agreement between pairs of annotators, A/B means we are looking at the agreement between the annotator A and annotator B

Secondly, the NaijaSynCor project has a central typological component, and language comparisons should be possible, based on syntactic differences, which is easier in a scheme based purely on distributional criteria, such as SUD, than on the rather semantic function word vs content word distinction that constitutes the basis of UD.

We can add that UD is particularly problematic for multi-words expression (MWEs) working as functional items (complex adpositions or complex conjunctions), especially when they are syntactically quite regular (Kahane et al., 2018). In SUD, MWEs such as the Naija adposition *base on* ‘(based) on’⁷ are connected and in the dependency tree they occupy the same syntactic position as a simple word, see (4).

- (4) but dat Love Wantintin⁸ < na base on wetin me dey see everyday //
‘but that Love Wantintin, it is based on what I see everyday’ [WAZK_11_M_Chiagozies-Life-Story_21]



The Naija treebank uses the SUD version proposed in (Gerdes et al., 2018), which is automatically converted into UD. Contrary to UD, two elements that are mutually exclusive and thus occupy the same syntactic position are linked to their governor by the same relation: For instance, *the problem* and *you’re wrong* are both **comp:obj** in *I know the problem* and *I know you’re wrong*, while the first is **obj** and the second is **ccomp** in UD. Considering that most of our readers are more or less familiar with UD, we choose to explain the specific SUD relations and how they are converted into UD. Adpositions (ADP) and subordinating conjunctions (SCONJ) govern the complement they introduce by the relation **comp**. These relations are reversed in UD: ADP **comp**> NOUN becomes NOUN **case**> ADP in UD and SCONJ **comp**> VERB becomes VERB **mark**> SCONJ.

For dependents of verbs, we distinguish between subjects (**subj**), complements (**comp**) and modifiers (**mod**). The relation **subj** becomes **nsubj** or **csbj** in UD according to the POS of the dependent. The relation **mod** becomes **advmod** for adverbs, **obl** for prepositional phrases, or **advcl** for clauses in UD. For verb complements, we distinguish the following sub-relations:

- **comp:obj**, for direct objects, which, in UD, becomes **obj** for a nominal dependent and **ccomp** for a clausal dependent;
- **comp:obl**, for oblique complements, which becomes **ccomp** for a verbal or clausal dependent, **iobj** for a nominal (or pronominal) dependent, and **obl** in other cases;
- **comp:pred**, for relations between two predicates that share an argument. This relations generally corresponds to UD’s **xcomp**⁹ but is reversed when the governor is a copula (AUX): AUX **comp:pred**> VERB becomes VERB **cop**> AUX in UD.
- **comp:aux**, for relations between a TAM (Tense–Aspect–Mood) auxiliary and the full verb, which is also reversed in UD.
- **compound:svc** is used for serial verb constructions, which are typical for Naija (see Section 3.3).

The difference between UD and SUD annotations is exemplified in Figure 2.

- (5) dem go seize am //

⁷ “base on” is not a passive construction in Naija as there is not morphological passive.

⁸ *Love Wantintin* is the name of a radio programme.

⁹ As remarked by (Przepiórkowski and Patejuk, 2018) and (Gerdes et al., 2018), raising is orthogonal to the syntactic function and it would be better to add **...:pred** to the syntactic function in case of raising, which would give us **comp:obj:pred** for objects with raising, **comp:obl:pred** for obliques with raising and **mod:pred** for modifiers with raising (such as *without talking* in *She explained it without talking*).

'They will seize it.' [DEU_C01_D_6]

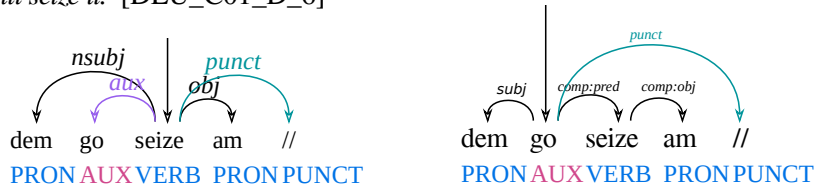


Figure 2. UD analysis vs. SUD analysis of (5)

All samples are first annotated by a trained annotator and the resulting trees together with the POS tags are then validated by an expert. Difficult cases are discussed among the annotators and the shared annotation guide is constantly updated. We apply simple error mining techniques such as looking for inconsistencies between the dictionary and the treebank. The SUD annotation scheme is a still ongoing process, and some special needs for the annotation of Naija have provided input for improvement of SUD.

2.5 Evaluation of treebank coherence and the impact of macrosyntactic annotation

In this section we will present the results of the annotation by evaluating parser performance on the current state of the treebank. In particular, we evaluate the relevance of macrosyntactic markup for syntactic parsing. We expect the macrosyntactic annotation to have a positive influence on dependency parsing, in particular for constructions such as coordination and dislocation, which have specific macrosyntactic markups resulting in specific dependency relations.

In order to verify this claim, we trained the Mate tagger and parser by Bohnet (2010), first on a version of the treebank with these markups, and then on a version of the treebank where they have been removed except for “//” (the segmentation into illocutionary units \approx sentences). This type of experiment is also important to set a baseline for the development of a Naija parser, to be used for parsing the rest of the NSC corpus (which is transcribed and macrosyntactically annotated) as well as for parsing other spoken and written data without markup.

We used a sample of 52k words, with 90% training and 10% test data on the Mate parser. While the POS tagging scores are as expected very similar whether macrosyntactic annotation is present or not, we obtain an LAS error reduction of 11% and a UAS error reduction of 18% through macrosyntactic annotation, see Table 1.¹⁰

	Macro-syntax +	Macro-syntax -	Error reduction
UPOS	92.44	92.23	*
UAS	90.76	89.23	18%
LAS	84.45	82.02	11%

Table 2. Parsing results with and without macro-syntax annotation.

Unsurprisingly, the syntactic functions which most benefit from this type of markup are those that are targeted by the annotation, such as piles (paradigmatic relations like **conj:coord**, **conj:dicto**, **compound:redup**) and coordinators (**cc**). We also observe an improvement for relations that connect a nucleus and adnuclei, such as clefts, dislocations, and peripheric modifiers.

The parser scores are promising,¹¹ in particular for spoken texts, and we hope to further improve the parser performance by the ongoing process of semi-automatic rule-based enhancement of the treebank coherence. In particular, we address this problem of annotation inconsistencies by a systematic comparison of parsing results with the gold annotation and the double SUD-UD-SUD conversion, and by different error mining tools such as the relation table proposed by the grew tool available on match.grew.fr (Bonfante et al., 2018), which shows the number of dependency relation types between any pair of categories. Also, the move to a neural network-based parser can be expected to result in better scores.

¹⁰ **punct** relations are excluded from the evaluation as they are exclusively used for macro-syntactic markers.

¹¹ Although Naija was part of the CoNLL 2018 Shared Task (Multilingual Parsing from Raw Text to Universal Dependencies), it is difficult to compare the results as Naija was one of the low-resource languages. The best score for UPOS, UAS, and LAS are 67.93, 38.62 and 30.07 (Zeman et al., 2018).

3 Some idiosyncratic syntactic constructions of Naija

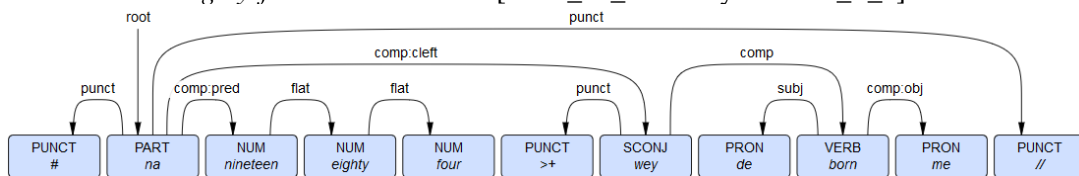
In this section we present three interesting constructions of Naija that show clear structural differences with English, Naija’s lexifying language .

3.1 Na-clefts and modifying relative clauses

Surface-syntax UD nicely captures the complexity of clefts¹² in Naija and the way they contrast with modifying relative clauses. We will restrict our presentation to one of the three cleft structures of Naija, *wey*-clefts.

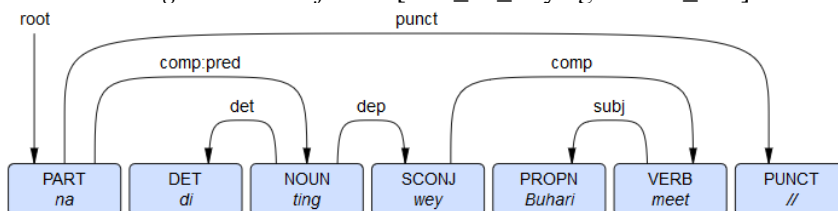
In Naija, clefts use the copula *na*,¹³ that has two complements: First, a *predicative complement*, linked by the relation **comp:pred** (*na* → *nineteen eighty-four* in (6)) and second, a clause introduced by *wey*, that we link to *na* with the **comp:cleft** relation (*na* → *wey de born me* in (6)). In the macrosyntactic markup, we use “>+” before the cleft clause.

- (6) # na nineteen eighty four >+ wey de born me //
 # it’s 1984 >+ that they bare me
 ‘it’s in nineteen eighty-four that I was born’ [KAD_09_Kabir-Gymnasium_P_6]¹⁴



Cleft sentences are superficially similar to copular predications in which the relative clause modifies the predicative complement. Yet, in clefts, the relation between the antecedent and the cleft clause is mediated by the copula, and the cleft clause is not dependent on the predicative complement but is raised and attached to the copula; whereas copular predications are thetic sentences that have a copula, a nominal, and a relative clause, but no syntactic restructuring and no backgrounding of the relative clause. The thetic meaning is clear when *na* has a presentational and not an identificational meaning, see example (7). In the syntactic representation of modifying relative clauses in copular non-identifying clauses, the copula takes only one complement: **comp:pred** (*na* → *di ting wey Buhari meet*):

- (7) na di ting wey Buhari meet //
 ‘this is the thing that Buhari found’ [IBA_25_Buying-Indomi_159]



3.2 Interrogatives

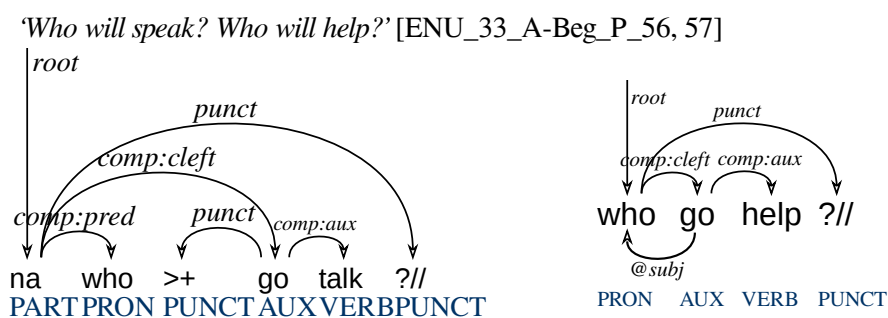
In the NSC corpus, content questions are analyzed as clefts. This is corroborated by examples where the question word of a content question can be preceded by the copular particle *na* without changing the behaviour or meaning of the sentence. The following two questions (8) occur in direct sequence and show how the copula *na* can be present or not without semantic consequence:

- (8) **na** who >+ go talk ?// who go help ?//

¹² Clefts are defined by Lambrecht (2001) as “a complex sentence structure consisting of a matrix clause headed by a copula and a relative or relative-like clause whose relativized argument is co-indexed with the predicative argument of the copula. Taken together, the matrix and the relative express a logically simple proposition, which can also be expressed in the form of a single clause without a change in truth conditions”.

¹³ *na* is classified as a particle and not a verb or an auxiliary in Naija because it cannot be negated or combined with TAM markers, two of the defining features of (auxiliary) verbs.

¹⁴ The NaijaSynCor project, entirely based on oral data, intends to study, among others, the interface between prosody and syntax. The # stands for a pause in the speech unit, a major cue for the study of prosody. The # is identified as a punctuation mark (PUNCT) in the syntactic representation.



This leads us to interpret question-words as focused, and the rest of the sentence as the focus-frame. In the absence of the focus particle *na*, the question word becomes promoted to **root** of the sentence through deletion of its previous head. In this analysis, the question word has a double function: It is the root of the sentence and a dependent of the verb.

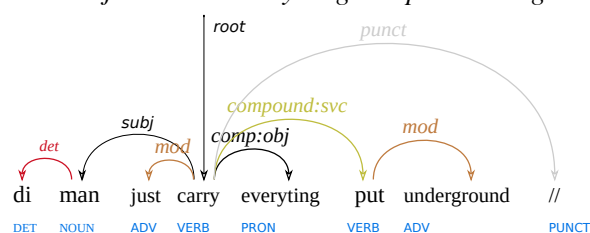
The complexity of the cleft structure of content questions cannot be captured by UD which treats the sentence verb as a root. Moreover, the parallelism between the two questions will not be kept by UD, as the one with *na* will be treated as a cleft, with the cleft phrase as the head, contrary to the other question without *na*. As a compromise between surface syntax and convertibility to UD, a second link has been added to the root, which annotates explicitly the dependency of the question word (this second relation is preceded by a “@”, see @subj in (8)). In our Surface-syntactic representation, both cases are represented by means of a cleft structure, see the above analysis. This is congruent with many analyses of *wh*-words which consider that they occupy two syntactic positions, one as a complementizer and another as a pronoun inside the clause they complementize (see, in particular, (Tesnière, 1959[2015]: ch. 246)).

During the conversion into UD we can only keep one of the relations, we have to keep the second relation as this follows the UD analysis of relative clauses. This leads to a “catastrophe” between the two syntactically related interrogative constructions (Gerdes and Kahane, 2016).

3.3 Serial Verb Constructions

The influence of adstrate vernacular languages, belonging mainly to the Niger-Congo family, is observed in the use of Serial Verb Constructions, that is “monoclausal construction[s] consisting of multiple independent verbs with no element linking them and with no predicate-argument relation between the verbs.” (Haspelmath, 2016). We used the subtyped relation **compound:svc** for these constructions. Sentence (8) contains an example of a serial verb construction (*carry* → *put*).

- (9) di man [...] just carry everyting put underground //
 'The man just carried everything and put it underground.' [ABJ_INF_12_Evictions_P_13]



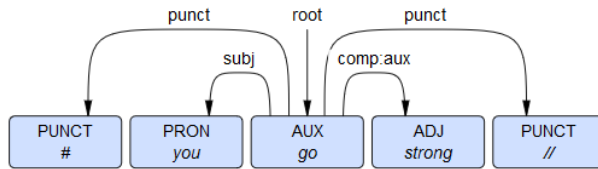
3.4 Polycategoriality and polyfunctionality

Following the UD guidelines¹⁵, the morphological specification of a (syntactic) word in the UD scheme consists of three levels of representation: a lemma, a POS tag, and a set of features representing lexical and grammatical properties. In order to reduce polycategoriality and its consequent multiplication of syntactic words, the annotation process has been guided by the principle of separation of the morphological tagging of a word from its syntactic function: A single lexeme can be polyfunctional, but it cannot be polycategorial. This principle applies in all languages, e.g. to adpositions (ADP) which can take a nominal, clausal or zero complement without changing their abstract lexical category (Huddleston and Pullum, 2008).

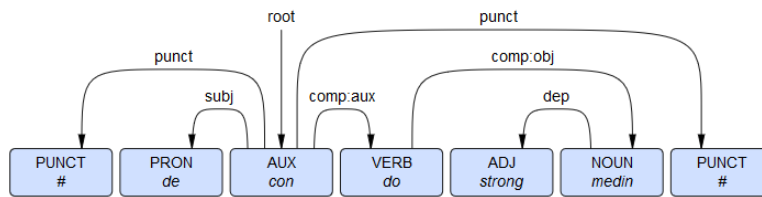
¹⁵ <https://universaldependencies.org/u/overview/morphology.html>

This principle has been applied to adjectives in Naija, which can function as predicates without any copula (10) or noun modifiers (11). In both cases, the words keep their morphological assignment: they are ADJ.

- (10) # you go **strong** //
 'you will (be) strong' [PRT_05_Ghetto-life_P_24]

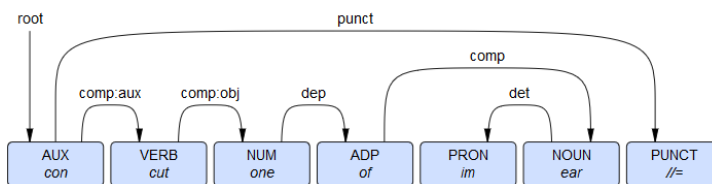


- (11) # de con do **strong** medin #
 'they then did strong magic' [IBA_04_Alaska-Pepe_P_95]

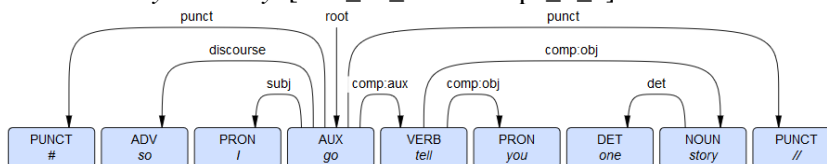


However, some lexical words are grammaticalized into new function words. An example is given by the numeral *one*, tagged NUM (12), which has grammaticalized into the determiner *one* 'some, a certain' (13), tagged DET, and the pronoun *one*, tagged PRON (14).

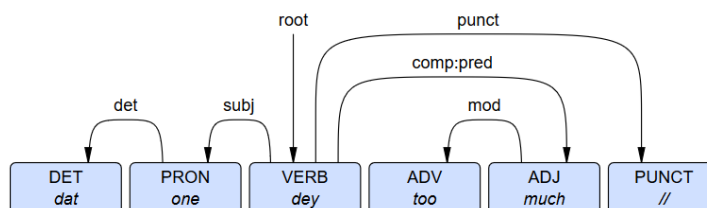
- (12) con cut **one** of im ear //=
 'he then cut one of its ears' [IBA_04_Alaska-Pepe_P_168]



- (13) # so I go tell you **one** story //
 'so I will tell you a story' [IBA_04_Alaska-Pepe_P_5]



- (14) dat **one** dey too much //
 'that one is too much' [ABJ_INF_08_Impatience_106]



3.5 Naija Grammar

The preliminary assessment of the NSC corpus has proved two things. First, despite the diversity of its speakers in terms of geographic origin and mother tongues, the corpus is remarkably homogeneous. Second, this homogeneity takes place while distancing the language from Nigerian Pidgin. Not only is new vo-

cabulary acquired through the necessity to cope with new functions and new cultures, but new grammatical structures are emerging and a new stability is found in the use of competing structures.

The study of Naija clefts is a good indicator. Naija clefts have three variants: **wey-clefts**, with a relative clause introduced by the relativizer *wey* (Section 3.1, example (5)), **bare clefts**, where the relativizer is omitted, resulting in a bare relative clause and **double clefts**, where the relativizer *wey* is replaced by a repetition of the copula followed by an expletive invariable 3sg pronoun: *na im*. They are exemplified in Table 1.¹⁶

1b'	wey-cleft	<i>na weekend wey we dey do am</i>	
1b''	bare cleft	<i>na weekend Ø we dey do am</i>	‘It’s in the weekend that we do it.’
1b'''	double cleft	<i>na weekend na im we dey do am</i>	

Table 3. The three structures of Naija clefts

We have quantified the relative use of these structures in Naija in a sub-section of 9621 sentences (almost 150 000 tokens) that constitute the syntactic treebank mirroring the social and geographic sampling of the full corpus, and compared those figures with Faraclas (2013), a presentation of the structures of Nigerian Pidgin with good data analysis. Using our own terminology, Faraclas’s figures highlight 3 main patterns representing fairly evenly cleft constructions in NP: *wey-clefts* (41%); bare clefts (39%) and zero-copula clefts (17%). Our own figures are respectively 1%, 89%, and 1%, with the rest of cleft patterns taken up by double clefts (9%). This shows a tendency in Naija, over the past 30 years, to marginalize *wey-* and zero-copula clefts, in favor of bare clefts, and give birth to a new pattern absent in Faraclas’s description, called double cleft, which seems to replace *wey-clefts*. In the double cleft construction, an emerging relative pronoun (*na im* → [nãĩ/nã] ‘who, which’) which is used only in this construction, replaces the relativizer *wey*, which is becoming specialized in modifying relative clauses.

4 Conclusion

We have described the workflow for the development of the gold section of the NSC treebank in the SUD annotation scheme, and we have shown the SUD analysis of some interesting syntactic constructions of Naija.

In parallel with the treebank construction we develop various interfaces to access the audio corpus, the transcription, and the different annotations. For example the most recent version of the SUD syntactic annotation is accessible at match.grew.fr/?corpus=SUD_Naija-NSC@dev.

In order to be part of the UD treebank family, an automatically converted UD version of the treebank will also be distributed, although the current UD platform does not foresee the joint distribution of the audio data. The increasing interest in spoken data will surely bring the UD community to discuss the format that will best allow for phonosyntactic studies. We will also distribute two text versions, one with macrosyntactic markup and a second version without the markup that can be used to train parsers on bare texts.

The perspective of this treebank creation goes beyond purely linguistic interest. It has deep sociolinguistic implications through the creation of a Naija dictionary. In order to create this treebank, we had to create an inventory of spelling variants, and we propose systematic distinctions of function and content words. The tools and resources of the NSC treebank enhances the interest in the specificity of Naija grammar, and the project can be seen as a step in the further establishment of Naija as a language (Courtin et al., 2018).

Acknowledgments

This research is financed by the French National Research Agency via the project ANR NaijaSynCor. We would like to express our gratitude to the Nigerian annotators: Onwueqbuza Emeka Felix, Ajede Chika Kennedy, and Tella Sansom Adekunle. The quality of the treebank has been largely enhanced by the constant interaction with Bruno Guillaume, the developer of the Grew platform. Thanks also to the anonymous reviewers of the SyntaxFest 2019 that helped us to clarify this paper.

References

Peter Bakker. 2009. Pidgins versus Creoles and Pidgincreoles. *The Handbook of Pidgin and Creole Studies*, John Wiley & Sons, 130–157.

¹⁶ See (Caron, 2019) for a complete presentation of clefts in Naija.

- Bernd Bohnet. 2010. Very High Accuracy and Fast Dependency Parsing is not a Contradiction. *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, Beijing, China.
- Guillaume Bonfante, Bruno Guillaume, Guy Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*. John Wiley & Sons.
- Bernard Caron. 2017. *NaijaSynCor: A corpus-based macro-syntactic study of Naija (Nigerian Pidgin)*. naijasyn-cor.huma-num.fr (September 2017).
- Bernard Caron. 2019. Clefts in Naija, a Nigerian pidgincreole. *Linguistics Discovery*, 41p.
- Christian Chanard. 2014. *ELAN-CorpA-V4.7.3*. http://llacan.vjf.cnrs.fr/res_ELAN-CorpA.php.
- Marine Courtin, Bernard Caron, Kim Gerdes, Sylvain Kahane. 2018. Establishing a Language by Annotating a Corpus. *Proceedings of the Workshop on Annotation in Digital Humanities (annDH)*, Sofia, ceur-ws.org/Vol-2155, 7-11.
- Emanuela Cresti. 2000. *Corpus di italiano parlato*. Accademia della Crusca, Florence.
- Lisbeth Degand, Anne-Catherine Simon. 2009. On identifying basic discourse units in speech: theoretical and empirical issues. *Discours*, 4, discours.revues.org.
- Henri-José Deulofeu, Lucie Dufort, Kim Gerdes, Sylvain Kahane, Paola Pietrandrea. 2010. Depends on what the french say: Spoken corpus annotation with and beyond syntactic function, *Proceedings of the 4th Linguistic Annotation Workshop (LAW IV)*.
- Dagmar Deuber. 2005. *Nigerian Pidgin in Lagos: Language contact, variation and change in an African urban setting*. Battlebridge Publications.
- Ben Ohiomamhe Elugbe, Augusta Phil Omamor. 1991. *Nigerian Pidgin: background and prospects*. Ibadan: Heinemann Educational Books Nigeria PLC.
- Nicholas Faraclas. 1989. *A grammar of Nigerian Pidgin*. PhD thesis, University of California at Berkeley.
- Nicholas Faraclas. 2013. Nigerian Pidgin structure dataset. In Susanne Maria Michaelis, Philippe Maurer, Martin Haspelmath & Magnus Huber (eds.), *Atlas of Pidgin and Creole Language Structures Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology. <https://apics-online.info/contributions/17>.
- Kim Gerdes, Sylvain Kahane. 2009. Speaking in piles: Paradigmatic annotation of french spoken corpus. *Proceedings of the Fifth Corpus Linguistics Conference*, Liverpool.
- Kim Gerdes, Sylvain Kahane. 2016. Dependency annotation choices: Assessing theoretical and practical issues of universal dependencies. *Proceedings of the 10th Linguistic Annotation Workshop (LAW-X)*, ACL, 131-140.
- Kim Gerdes, Bruno Guillaume, Guy Perrier, Sylvain Kahane. 2018. SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD, *Proceedings of the Universal Dependencies Workshop (UDW)*, EMNLP, Bruxelles.
- Kim Gerdes, Bruno Guillaume, Guy Perrier, Sylvain Kahane. 2019. Improving Surface-syntactic Universal Dependencies (SUD): surface-syntactic relations and deep syntactic features, *Proceedings of the Universal Dependencies Workshop (UDW)*, SyntaxFest, Paris.
- Rodney Huddleston, Geoffrey K. Pullum. 2008. *The Cambridge Grammar of the English Language* (2nd ed.), Cambridge: Cambridge University Press.
- Sylvain Kahane, Marine Courtin, Kim Gerdes. 2018. Multi-word annotation in syntactic treebanks: Propositions for Universal Dependencies, *Proceedings of the 16th international conference on Treebanks and Linguistic Theories (TLT)*, Prague.
- Knud Lambrecht. 2001. A framework for the analysis of cleft constructions. *Linguistics* 39(3). 463–516.
- Luigi (Yu-Cheng) Liu, Anne Lacheret-Dujour, Nicolas Obin. 2019. Automatic Modelling and labelling off speech prosody: What's new with SLAM+?. *Proceeding of the International Congress of Phonetic Sciences (ICPhS)*, Melbourne.
- Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, et al. 2018. *Universal Dependencies 2.2*. <https://hal.archives-ouvertes.fr/hal-01930733>.
- Timothy Osborne, 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).
- Paola Pietrandrea, Sylvain Kahane. 2019. The macrosyntactic annotation. In Anne Lacheret-Dujour, Sylvain Kahane, Paola Pietrandrea (eds.), *Rhapsodie: A Prosodic and Syntactic Treebank for Spoken French*, John Benjamins, Amsterdam. 97-126.

- Adam Przepiórkowski, Agnieszka Patejuk. 2018. Arguments and adjuncts in Universal Dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, Santa Fe, NM, 3837–3852.
- Han Sloetjes, Peter Wittenburg. (2008). Annotation by category – ELAN and ISO DCR. *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*
- Sali Tagliamonte. 2012. *Variationist sociolinguistics: change, observation, interpretation* [Language in Society, 40]. Malden, MA: Wiley-Blackwell.
- Isabelle Tellier, Iris Eshkol, Samer Taalab, Jean-Philippe Prost. 2010. POS-tagging for oral texts with CRF and category decomposition. *Research in Computing Science*, 46, 79-90.
- Lucien Tesnière. 1959. *Éléments de syntaxe structurale*. Paris: Klincksieck. [Transl. by Timothy Osborne, Sylvain Kahane, 2015. *Elements of structural syntax*, Amsterdam: John Benjamins.]
- Peter Withers. 2012. Metadata Management with Arbil. In Victoria Arrantz, Dan Broeder, Bertrand Gaiffe, Maria Gavrilidou & Monica Monachini (eds.), *Proceedings of the workshop Describing LRs with Metadata: Towards Flexibility and Interoperability in the Documentation of LR*, LREC, 72–75.
- Daniel Zeman et al. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.

Can Greenbergian universals be induced from language networks?

Kartik Sharma¹, Kaivalya Swami¹, Aditya Shete², and Samar Husain³

¹ Department of Computer Science & Engineering, Indian Institute of Technology Delhi

² Department of Physics, Indian Institute of Technology Delhi

³ Department of Humanities & Social Sciences, Indian Institute of Technology Delhi

cs1170342@iitd.ac.in, kaivu1999@gmail.com,

adityashete009@gmail.com, samar@hss.iitd.ac.in

Abstract

Language networks have been proposed to be the underlying representation for syntactic knowledge (Roelofs, 1992; Pickering and Branigan, 1998). Such networks are known to explain various word order related priming effects in psycholinguistics. Under the assumption that word order information is encoded in these networks, we explore if Greenbergian word order universals (Greenberg, 1963) can be induced from such networks. Language networks for 34 languages were constructed from the Universal Dependencies Treebank (Nivre et al., 2016) based on the assumptions in Roelofs (1992); Pickering and Branigan (1998). We conducted a series of experiments to investigate if certain network parameters can be used to cluster various languages based on the word order typology proposed by Greenberg. Our results show that some network parameters robustly cluster the languages correctly, thereby providing some support for language network as a valid representation for such linguistic generalizations.

1 Introduction

Establishing connections and relations between objects is an important way of representing knowledge (Siew et al., 2018). Such a representation lends itself to a succinct understanding of its structure and complexity. Such network representations are routinely used to understand complex systems such as social systems, biological systems, economic systems and so on (Pastor-Satorras and Vespignani, 2007; Caldarelli, 2007; Newman, 2010). Language seems well suited for this type of representation; after all, the knowledge of language and its use, is primarily about establishing relations between different kinds of linguistic objects (Borge-Holthoefer and Arenas, 2010; Solé et al., 2010). Indeed, the significance of such networks was appreciated quite early in the domain of meaning representation in terms of semantic relatedness (Collins and Loftus, 1975; Ober and Shenaut, 2006). Such semantic networks have been shown to capture experimental results on lexical priming (McRae and Boisvert, 1998; McRae et al., 2005). Additionally, various resources (e.g., Wordnet) and as well as models (e.g., word2vec) have been proposed with the motivation of establishing relations between similar words (Miller, 1995; Mikolov et al., 2013). A network-based representation has also been proposed to subserve syntactic knowledge in the mind (Roelofs, 1992; Pickering and Branigan, 1998). Such a network has been claimed to correctly explain the syntactic priming effects in during language production/comprehension (Pickering and Ferreira, 2008; Tooley and Traxler, 2010). Networks have also been used to quantify cognitive processes and representations related to various linguistic levels such as words, etc. (e.g Vitevitch, 2008; Allegrini et al., 2004; Chung and Pennebaker, 2007; Morrill, 2000; Vitevitch et al., 2011).

Network theory has been extensively used to understand (and visualize) such knowledge representations (Barabási, 2011). Network theory formalizes a knowledge system as a network, which contains nodes and edges describing the entities and the relations between them. Network theory enables us to extract specific information related to the connectedness and relationships between various entities (Newman, 2010; Costa et al., 2011). The primary attraction of representing a complex system in the form of a network lies in the ease with which various relations present in the data can be visualized. In addition, it has the ability to abstract the relations at different levels, ranging from a single node, to viewing the properties of the entire network as a whole (Albert et al., 2000).

The idea of language as a network has been gaining some traction in computational linguistics (e.g., Ferrer-i Cancho et al., 2007; Lerner et al., 2009; Ke and Yao, 2008; Borge-Holthoefer and Arenas, 2010; Lerner et al., 2009; Choudhury et al., 2010; Ferrer-i Cancho and Solé, 2001; Vitevitch et al., 2011; Ferrer-i Cancho et al., 2004; Čech et al., 2011; Liu and Xu, 2011; Mehler et al., 2016). One approach, that we explore here, is to construct language networks from annotated dependency treebank to encode syntactic relationship between lexical items. Previous works on such language representation have explored the properties of language networks formed through dependency treebanks (Ferrer-i Cancho et al., 2004), also see Cong and Liu (2014). Relatedly, Liu and Li (2010); Abramov and Mehler (2011) used language network to successfully cluster languages into phylogenetic groups using network parameters. As stated earlier, networks have also been hypothesized to be the representation that subserves syntactic knowledge in the mind (Roelofs, 1992; Pickering and Branigan, 1998). In particular, it has been used to explain syntactic priming with respect to various word order choices during sentence comprehension and production (Pickering and Ferreira, 2008; Tooley and Traxler, 2010). This implies that networks can represent various syntactic rules (e.g., word order) in terms of nodes and their relationship with other nodes in the network. In other words, the network as a representation of language should contain the same generalisations as present in a language. Greenberg’s universals (Greenberg, 1963) are a set of such generalisations that occur across languages. These universals and their status in language networks is the focus of this article.

In this work¹, we build a psycholinguistically motivated language network (Roelofs, 1992; Pickering and Branigan, 1998) for 34 languages to investigate if Greenberg’s word order related language universals (GU) can be induced from the networks. To do this, we conduct two experiments. In the first experiment, we simply map the GUs onto a language network to see if a particular node property (percentage of outgoing arcs) leads to the desired classification across languages. For example, for GU universal no. 3, we look at this parameter of the VSO nodes across all language networks and see if the parameter values cluster the respective languages as prepositional or postpositional. In the second experiment, we automatically derive certain implicational universals stated by Greenberg. For example, we see which word order node (e.g., SVO, SOV, etc) best classifies the order of adposition and noun phrase. In effect, the first experiment is completely correlational and supervised – checking **if** a known node parameter leads to the correct language typology. The second experiment, is unsupervised – checking **which** node (and its parameter) leads to the correct language typology. Together, the two experiments shed light on whether language network can induce correct GU wrt word order and highlights the properties of the network where this information can be found.

The paper is arranged as follows. We begin with a description of the data, tools and network formation in the Section 2. In section 3 we present the two experiments and discuss the results. Following this, in section 4 we conclude the paper and list out some future directions.

2 Methodology

2.1 Data and Tools

We use the ‘**Universal Dependencies**’ Treebank (UD for short, henceforth) (Nivre et al., 2016; Agić et al., 2015) to create the network. The UD has annotated data for over 70 languages in the latest version, of which we are utilizing 34². Only those languages were selected that had a relatively large size (sentence count more than 2k) and that were present in the WALS (The World Atlas of Language Structure) database (Dryer and Haspelmath, 2013). WALS data in .csv format is directly available from the WALS online source. The UD CoNLL-U format was converted into a network (edges and nodes data) format in order to use the Cytoscape (Shannon et al., 2003) software. Cytoscape is an open-source network visualization and analysis software.

¹The data and the code (along with details about various calculations) used in this paper have been made available at https://github.com/Ksartik/SyntaxFest2019_paper18

²Ancient Greek, Arabic, Basque, Bulgarian, Catalan, Chinese, Croatian, Czech, Danish, Dutch, English, Estonian, French, German, Hebrew, Hindi, Indonesian, Italian, Japanese, Latvian, Norwegian, Persian, Polish, Portuguese, Romanian, Russian, Slovak, Slovene, Spanish, Swedish, Turkish, Ukrainian, Urdu

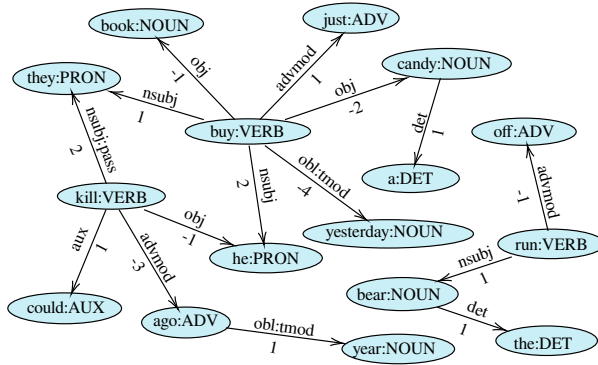


Figure 1: A sample base-network (see point 1 in Section 2.2) derived from 4 sentences. These sentence are ‘They could kill him years ago’, ‘He just bought a candy yesterday’, ‘The bear ran off’, ‘They buy books’.

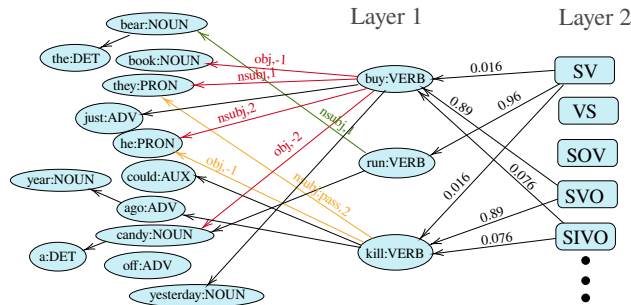


Figure 2: The final network derived from the base network above (Figure 1) for the same sentences. All reported results follow such a representation.

2.2 Language Network

The language network derived from the UD data is motivated by the syntactic representation proposed by Roelofs (1992) and adapted by Pickering and Branigan (1998). This model has been used to explain syntactic priming during comprehension (Pickering and Ferreira, 2008) and production (Tooley and Traxler, 2010). The model consists of layers of linguistic elements connected to each other. Nodes representing word tokens are connected to ‘lemma’ nodes. The ‘lemma’ nodes are associated with syntactic information such as category, morphological information, etc. The ‘lemma’ nodes associated with the verbs are connected to the ‘combinatorial’ nodes representing their syntactic subcategorization information, in other words, the typological word order information. When a verb is required in speech, an activation of a concept results in the selection of the highest activated ‘combinatorial’ node which in turn activates the relevant ‘lemma’ node. Interestingly, activation of this ‘lemma’ node results in the activation of syntactically similar verbs. This is because verb lemma that have similar syntactic properties are linked to the same combinatorial nodes. In this work, we construct a similar network. This results in a layered network in which the last layer explicitly contains word-order properties such as ‘SVO’, ‘SOV’, ‘VSO’, etc. The ‘combinatorial’ node described in the network discussed in Pickering and Branigan (1998), is modelled here as a node which encapsulates the argument structure of the verb nodes connected to it.

Creation of the network is done in multiple steps, which we describe below. We illustrate this through Figures 1 and 2 above.

1. Universal Dependencies Treebank data was converted to a node and edge data. The nodes are defined as the LEMMA of a word tagged with its part of speech (UPOS), which we will call LEMMA:UPOS. The other properties (e.g., FEATS) of each node given in the CoNLL-U format are also associated with each node. The edges between the nodes are directed and represent dependency links from HEAD of a word/node to the dependent node. In addition, the edges have certain attributes such as (a) linear distance: distance between the connected nodes based on the linear position of the nodes in the corresponding sentence (calculated as HEAD - INDEX from the CoNLL-U format), (b) dependency relation (DEPREL) : dependency relation between the nodes (provided as DEPREL in the CoNLL-U

format. The resulting network at this stage is shown in Figure 1.

2. Next, we select only those verb lemma nodes that are finite³ (obtained from VerbForm attribute in the FEATS column in the CoNLL-U data). This is done in order to have a more robust generalization regarding the argument structure of individual verbs as non-finite instances of verbs can drop their arguments. This leads to the formation of Layer 1 shown in Figure 2.
3. We then create layer 2 (see Figure 2) which has nodes corresponding to various word order possibilities of verb arguments, e.g., SOV, SVO, VSO, etc. These layer 2 ‘combinatorial’ nodes are connected to the layer 1 lemma nodes. The connection between the lemma and the combinatorial node represents the probability of a verb appearing with a particular argument structure and its word order. We considered all the combinations (without replacement) of ‘S’ (denoting **subject**), ‘V’ (denoting **verb**), ‘O’ (denoting **object**), ‘I’ (denoting **indirect object**) containing at least one ‘V’. Some of these nodes are : **SV, VS, SOV, SVO, SIOV** etc. Layer 2 thus consists of 48 pre-defined nodes⁴ similar to combinatorial nodes in Pickering and Branigan (1998).
 - These combinatorial nodes are obtained by computing two layer 1 properties. These are average sentential distance of the core arguments (subject, object and indirect object) and their proportions. Average sentential distance is obtained by grouping all the nodes with argument relation edges and computing their average linear distance from the verb. This is done for each core argument. In addition we also compute the proportion of each core argument in a group relative to total no. of core arguments for a verb in layer 1.
 - In order to connect the verb lemmas in layer 1 with the nodes in layer 2, we computed the probabilities with which these verbs appear in a specific argument structure configuration in the treebank. We assume that the word order of a certain verb remains same and it is just the argument structure that can show variations.⁵ The average distance of the verb relative to the argument can be formalized as a tuple (*subj-dist*; *obj-dist*, *inobj-dist*), where, *subj-dist* is the average distance between the verb and the subject group, etc. For example, if the distances are (1; -1; 0), then the word order is SVO. If the word order is SVO, the concerned verb can connect to any of the following - SV, SVO, SVIO, SIVO, SVOI, ISVO.
 - In order to identify which one of the above possibilities the verb must have, we devised probabilities for each possible node. Here, we used the proportionate size of each group - ‘subject’, ‘object’, ‘indirect object’, as a parameter to find the probability. For example, if the proportion is given as (0.5, 0.5, 0) then it is expected that the verb is transitive. On the other hand, a proportion of (0.75, 0.25, 0) does not clearly identify a certain group and thus we need a method to associate a verb with more than one group.⁶
4. We then formed layer 2 (as shown in the Figure 2) – connecting the verbs with the edges that have the probabilities as their weights. As discussed, layer 2 of the language network comprises of the ‘combinatorial’ nodes which are connected to the verb lemma nodes from layer 1 of the network. The combinatorial nodes store the argument structure as well as word order property of its connected nodes. The probabilities on the edges connecting these nodes to the lemmas denotes the weights of these connections. Considering Figure 2, the probabilities of connections of ”buy:VERB” (in Layer 1)

³The finiteness information is determined using both the FEATS of both verb lemma as well as its auxiliary. Also, note that this will give us both main and subordinate clauses. In this work we ignore the fact that some languages have different word order in main vs subordinate clause.

⁴Specifically, the 48 nodes are SV, VS, OV, VO, IV, VI, SVO, SOV, VSO, VOS, OVS, OSV, ISV, IVS, VSI, VIS, SIV, SVI, IOV, IVO, VOI, VIO, OVI, OIV, SIOV, SIVO, SOIV, SOVI, SVOI, SVIO, IOVS, IOSV, IVSO, IVOS, ISVO, ISOV, VOSI, VOIS, VISO, VIOS, VSIO, VSOI, OVIS, OVSI, OSIV, OSVI, OIVS, OISV

⁵In a way, capturing the dominant word order pattern of a verb which we are really interested in.

⁶We considered the proportions of subject, object, indirect object as a vector in 3D space. We have a pre-defined set of proportions (or classes) which correspond to the layer 2 nodes – (1,0,0): SV/VS, (0,1,0): VO/OV, (0.5, 0.5, 0): transitive of any order and so on. Since these nodes or target vectors are not distributed uniformly in terms of distance, we used the angular distance of the corresponding unit vectors as a measure to calculate probabilities (after proper normalization). This method allowed us to remove any bias for an input proportionate vector vis-à-vis a particular layer 2 node. More details regarding computation of probabilities can be found at https://github.com/Ksartik/SyntaxFest2019_paper18

with "SV" (0.016), "SVO" (0.89) and "SIVO" (0.076), shows that "buy" predominately follows "SVO". The sample network shown in figure 2 shows that the language that this network represents is a "SVO" language.

3 Experiments

The experiments discussed in this section assume that a large probability is related with a strong connection and more likelihood that the connected nodes show the 'combinatorial' property encapsulated by the concerned Layer 2 node. Further, in order to do the network analysis, we used the sentential distance only as a weight to the edges. For the connections between Layer 1 and Layer 2, we used the inverse of the probabilities as the edge weights so that the range is from $[1, \infty]$. All network analysis was performed using Cytoscape (Shannon et al., 2003). In particular, Cytoscape provides a tool named Network Analyzer which was used to analyse the network with various parameters⁷. All analysis reported below has been done on the network parameters corresponding to the nodes in layer 2.

In the first experiment, we simply map the GUs onto a language network to see if a particular node property (percentage of outgoing arc) leads to the desired classification across languages. In the second experiment, we automatically derive certain implicational universal stated by Greenberg (1963). For example, we see which word order node (e.g., SVO, SOV, etc) best classifies the order of adposition and noun phrase.

3.1 Experiment 1

In order to map the Greenbergian universals wrt certain linguistic orders onto the network, we reduced the problem to only probing the node parameters of the layer 2 'combinatorial' nodes. This was done because we are interested in word order generalizations related to the verb. In particular, we looked at each of the word-order based Greenbergian universal and translated them to a particular network parameter of various combinatorial nodes in layer 2. The orders SOV, SVO, VSO etc. are believed to be encoded in the parameter 'Outperc' of the layer 2 nodes. 'Outperc' is defined as the out-degree of the concerned node divided by the total no. of nodes in layer 2. A language is deemed to be SOV if the SOV node's 'Outperc' is high relative to other nodes in layer 2. We investigate if the distribution of 'Outperc' across all language networks leads to the correct language typology clusters. This experiment is intended as a supervised way of identifying language typology clusters based on Greenberg's word order universals. The data available in WALS (Dryer and Haspelmath, 2013) was used to get the word order patterns related to the Greenbergian universals for various language.

3.1.1 Results

Two network parameters, namely 'Outperc' and 'Outdegree' were used for analysis. As stated above, 'Outperc' is the fraction of verbs connected to a particular combinatorial node. 'Outdegree' is the number of verbs connected to a particular class. The results for various universals are given below

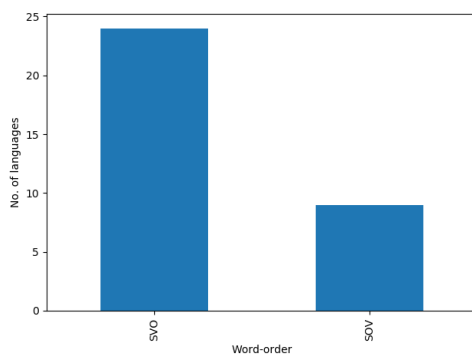


Figure 3: Dominant subject and object order across all language networks.

⁷These were, *In-degree*, *Out-degree*, *Outperc*, *Edge Count*, *Average shorted path length*, *Betweenness centrality*, *Closeness centrality*, *Closeness centrality*, *Clustering coefficient*, *Neighborhood connectivity*, *Eccentricity*. For details on these parameters, see Newman (2010). Also see: <https://med.bioinf.mpi-inf.mpg.de/netanalyzer/help/2.7/index.html#complex>

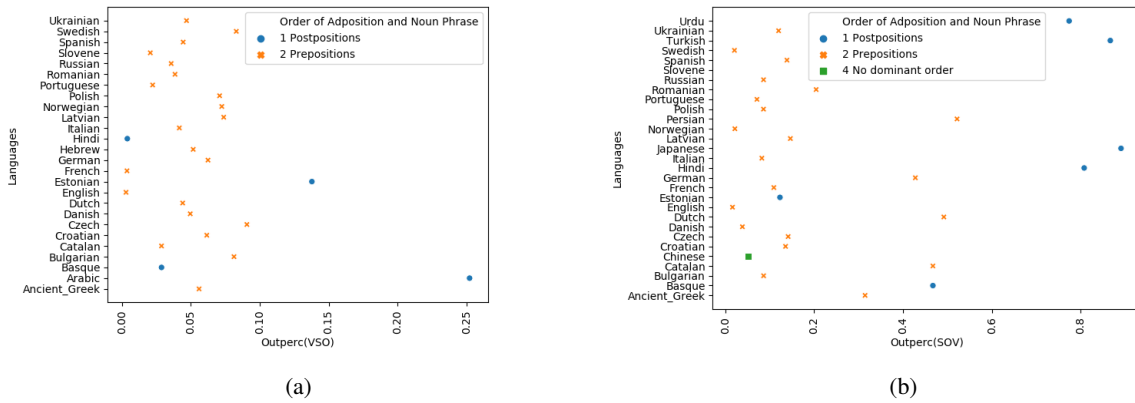


Figure 4: (a) Outperc for VSO node across all languages and corresponding typology clusters based on postpositional vs prepositional languages. (b) Outperc for SOV node across all languages and corresponding typology clusters based on the order of adposition and noun phrase.

1. Universal 1 - *“In declarative sentences with nominal subject and object, the dominant order is almost always one in which the subject precedes the object.”*

Since we used only finite verb forms in the second layer, the properties shown in the third layer are expected to be of a general rule for declarative sentences. Figure 3 shows the histogram of the maximum ‘Outperc’ over all 34 languages.

Results show that for all the languages, ‘Outperc’ is maximum for either SOV or SVO nodes. Thus, verifying the universal using the networks used in this analysis.

2. Universal 3 - *“Languages with dominant VSO order are always prepositional.”*

The feature “85A Order of Adposition and Noun Phrase” in WALS was used to get the information on languages with prepositional vs post-positional. As mentioned above, none of the language networks have a dominant VSO order. Nevertheless, we went ahead to form the clusters using the ‘Outperc’ of the VSO nodes. The clustering is shown in figure 4a.

Results show that a higher “VSO outperc” corresponds to post-positional feature. We conclude that our network is not able to induce this universal in its strong form. One reason for this could be that the none of the treebank data for the languages used (including Arabic) had a dominant VSO order for finite verbs.

3. Universal 4 - *“With overwhelmingly greater than chance frequency, languages with normal SOV order are post-positional.”*

Similar to the previous approach, we looked at the ‘Outperc’ of the SOV nodes in various language networks and looked at the resultant clustering. Figure 4b shows the clusters.

Results show a clear classification of languages with postpositions vs prepositions. We see that higher values of ‘Outperc’ for SOV nodes correspond to postpositional languages, with the exception of Estonian.

4. Universal 5 - *“If a language has dominant SOV order and the genitive follows the governing noun, then the adjective likewise follows the noun.”*

We looked at the languages where genitive follows the noun using the WALS data and then made the clusters of SOV node’s distribution based on the adjective-noun order. This is shown in figure 5a.

Results show - that both ‘Outperc’ as well as ‘Outdegree’ for the SOV nodes were not able to cluster the languages correctly.

5. Universal 6 - *“All languages with dominant VSO order have SVO as an alternative or as the only alternative basic order.”*

Since we didn’t have any language with dominant VSO order, we show a comparative plot of ‘Outperc’ of SVO and VSO across languages in figure 5b.

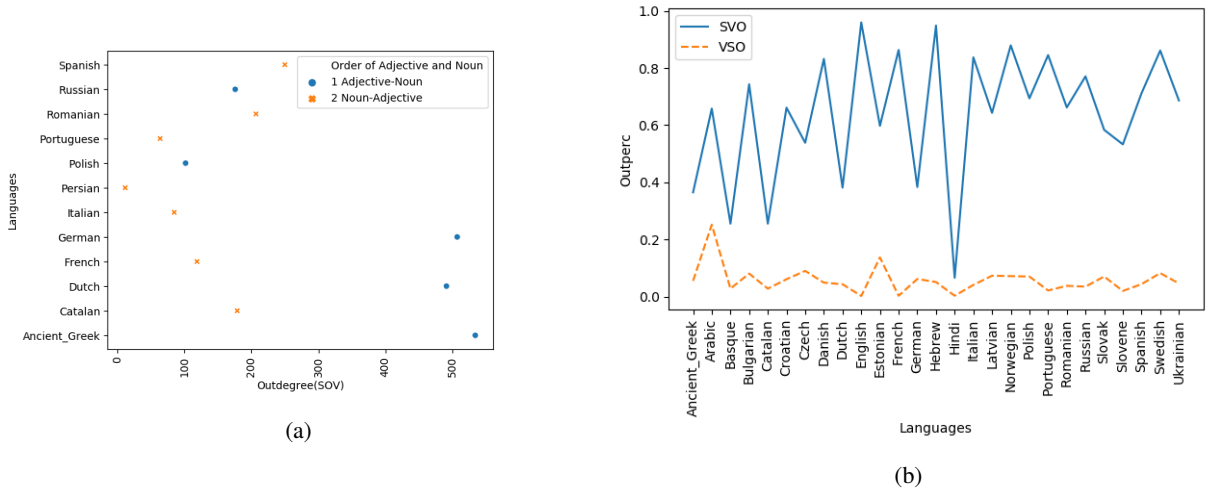


Figure 5: (a) Outdegree for SOV node across all languages (with genitive following nouns) and corresponding typology clusters based on order of adjective and noun. (b) Outperc values across various languages for VSO and SVO nodes.

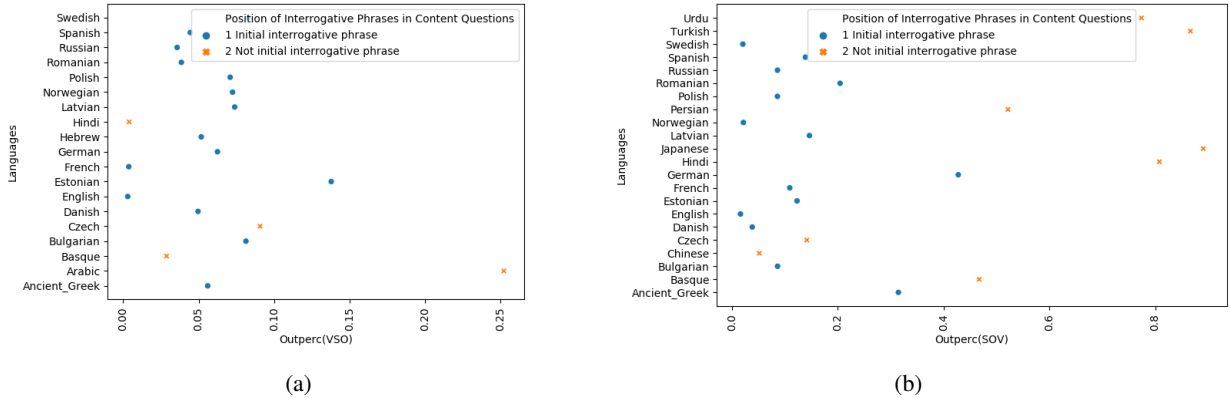


Figure 6: Outperc for VSO node across all languages and corresponding typology clusters based on the order of interrogative phrases.

A correlation analysis suggests that, other than certain languages, over all, the R^2 came out to be just 0.07, suggesting that the networks are unable to capture this generalization.

6. Universal 12 - *“If a language has dominant order VSO in declarative sentences, it always puts interrogative words or phrases first in interrogative word questions; if it has dominant order SOV in declarative sentences, there is never such an invariant rule.”*

We used the relevant feature in WALS data to plot the ‘Outperc’ of VSO and SOV for the languages obtained from WALS. This is shown in figure 6

Results show that increase in the VSO ‘Outperc’ does not lead to the right typology cluster. Interestingly, the ‘Outperc’ for SOV nodes for different languages gave better results. Thus providing partial support for the universal from the networks.

To summarize, the result show that the language typology related to (a) order of subject-object across languages, (b) presence of prepositions in SOV languages, and (c) position of interrogative word in VSO/SOV language, can be derived from the ‘Outperc’/‘Outdegree’ parameter of the layer 2 nodes in various language networks.

3.2 Experiment 2

Experiment 1 targetted a specific universal and mapped it on to the network using a prespecified node property (Outperc/Outdegree of SOV, VSO, SVO layer 2 nodes). In experiment 2, we asked a more general question – which node parameter in different language networks leads to the best language typology classi-

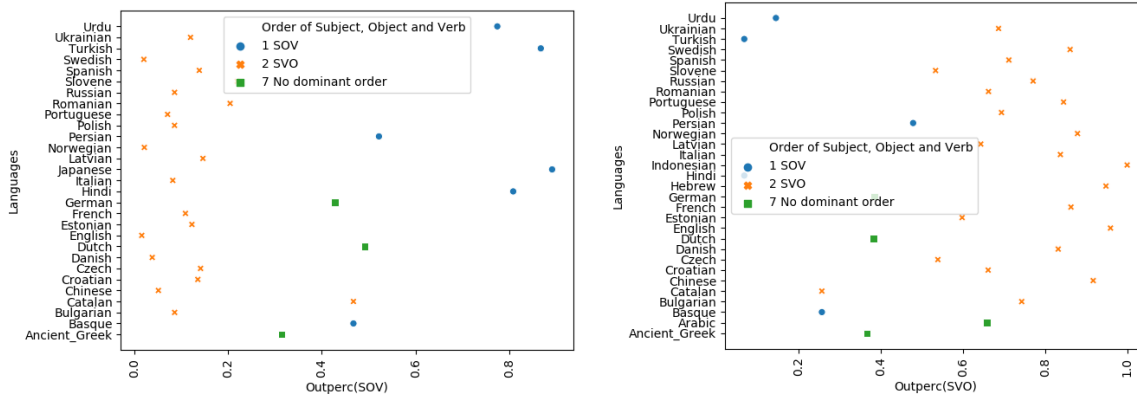


Figure 7: Top two language clusters wrt the order of subject, object and verb. The Outperc parameter for SOV nodes across all languages lead to the best distinction between SOV vs SVO languages.

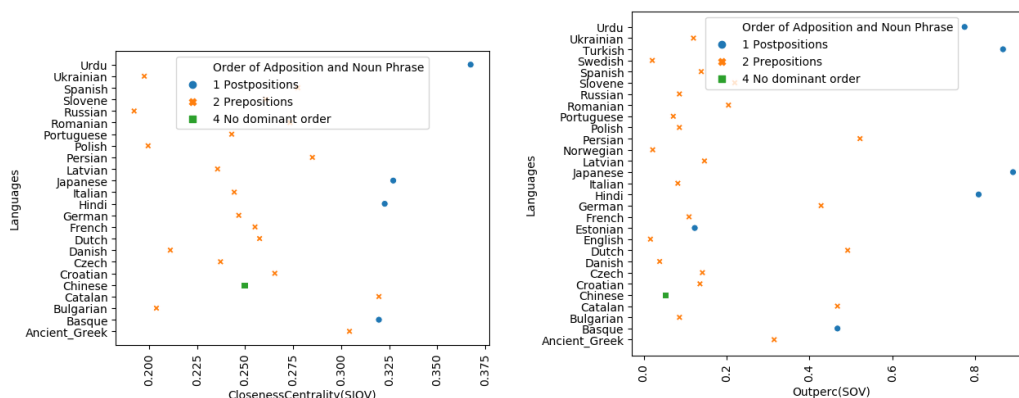


Figure 8: Manually identified language clusters wrt the order of adposition and noun phrases. The Outperc parameter for SOV nodes across all languages lead to a good distinction between language where the adposition follows the noun phrase vs those where it precedes the noun phrase.

fication based on Greenberg’s universals? The linguistic orders that we looked at were taken from WALS (Dryer and Haspelmath, 2013)]; these were, (a) Order of subject, verb and object, (b) Order of Adposition and Noun Phrase, (c) Order of Adjective and Noun, and (d) Position of Interrogative Phrase and Content Questions.

We investigate various parameters⁸ for each node in layer 2 to see which node-parameter combinations across all the languages lead to the best language classification for a particular word order. For example, consider “Order of Adposition and Noun Phrase”. In order to find which parameter of which layer 2 node can lead to the best classification of languages based on this order, we get a particular node-parameter values from all language networks, and check if this distribution leads to the correct classification of languages as given in the WALS data. The correlation between the node-parameter values and the correct language cluster (which is already known) is quantified by silhouette value (Rousseeuw, 1987). This silhouette value is obtained for all the (nodes \times parameters) node-parameter combinations and the highest score gives us the node-parameter that classifies the languages best based on the word order under consideration. A greater silhouette value corresponds to better clustering. Intuitively, the silhouette value captures the cohesiveness of the data point with its cluster.

To summarize, experiment 2 discusses a method to induce the linguistic orders by probing all possible parameters for each verb-order nodes that are contained in layer 2.

⁸These were, *In-degree*, *Out-degree*, *Outperc*, *Edge Count*, *Average shorted path length*, *Betweenness centrality*, *Closeness centrality*, *Closeness centrality*, *Clustering coefficient*, *Neighborhood connectivity*, *Eccentricity*. For details on these parameters, see Newman (2010). Also see: <https://med.bioinf.mpi-inf.mpg.de/netanalyzer/help/2.7/index.html#complex>

3.2.1 Results

Below we discuss the results for the various word order patterns. The top two clusters based on silhouette values are shown for each pattern.

1. Order of Subject, verb and object:

Results show that ‘Outperc’ of the SOV node clusters the languages much better than ‘Outperc’ of the SVO node (see Figure 7). Results also suggest that ‘Outperc’ outperforms all other node parameters. Recall that ‘Outperc’ is the percentage of outgoing edges from a node. This means that, as far as the current set of languages is considered, the ‘Outperc’ property of the ‘SOV’ node can alone be effectively used to decide the word order of the language. This suggests that there is a lot of variability wrt SVO order in various languages compared to SOV order.

2. Order of Adposition and Noun Phrase :

The top silhouette scores for various parameter-node pairs did not lead to a good cluster of languages based on this feature. This is not to say that the appropriate clustering cannot be derived from the cluster. Indeed, a manual analysis of the various clusters shows that the ‘ClosenessCentrality’ parameter of SIOV nodes across all the languages does lead to good language clusters for this feature. In addition, ‘Outperc’ of the SOV nodes leads to good clusters (see Figure 8). ‘ClosenessCentrality’ gives us a measure of how close the node in question is to the other nodes in the network. Given this definition, it is difficult to see why such a parameter should lead to the correct clustering. Interpreting the results on the other parameter, namely, ‘Outperc’ for SOV nodes is easier. It shows that the order of subject, object and verb can predict the order to adposition and noun phrase as was hypothesized by Greenberg.

3. Order of Adjective and Noun:

The two clusters based on silhouette scores show that ‘Neighborhood Connectivity’ of OVIS and VOSI nodes for various languages were able to cluster the languages really well (see Figure 9). ‘Neighborhood Connectivity’ corresponds to the average connectivity of its neighbours. While the result does give us the desired clusters, it is difficult to interpret the linguistic validity of the parameter.

4. Position of Interrogative Phrase in Content Questions :

Finally, for the cluster based on position of interrogative phrase the silhouette scores for the cluster based on “Outperc” parameter for the SOV nodes gave one of the best results (see Figure 10).

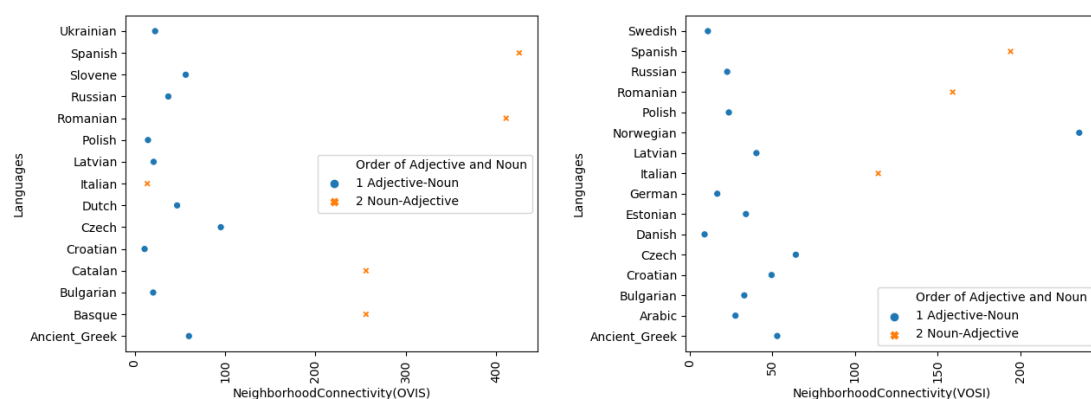


Figure 9: Top two language clusters wrt the order of adjective and noun.

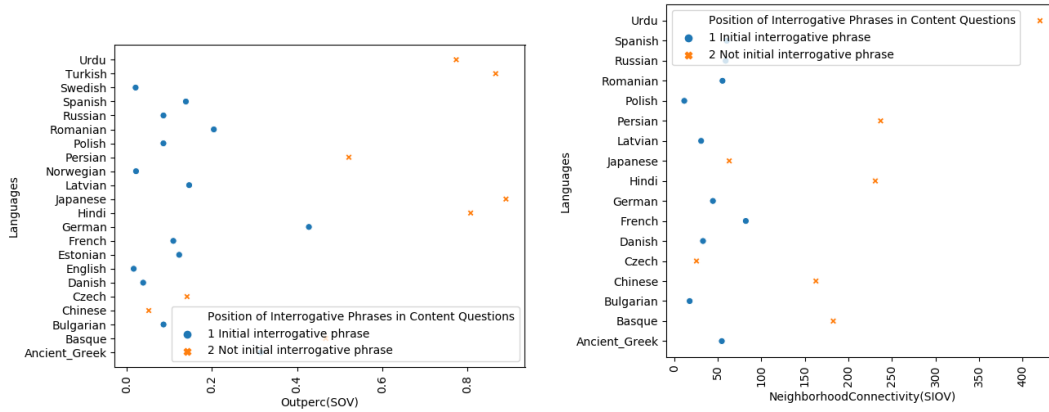


Figure 10: Top two language clusters wrt the position of interrogative phrase in content questions.

WALS feature	Network Parameter 1			Network Parameter 2			Network Parameter 3		
	Node	Parameter	Silhouette	Parameter	Parameter	Silhouette	Node	Parameter	Silhouette
81A	SOV	Outperc	0.53	SVO	Outperc	0.304	OSV	Outperc	0.3
85A	SIOV	Closeness C	-0.25	SOV	Outperc	-0.25	-	-	-
87A	OVIS	Neighborhood C	0.63	VOSI	Neighborhood C	0.58	OISV	Eccentricity	0.34
93A	SOV	Outperc	0.48	VOIS	Neighborhood C	0.604	SIOV	Neighborhood C	0.466

Table 1: Top 3 silhouette score for the clusters related to the 4 word order patterns. 81A: Order of subject, verb and object; 85A: Order of adposition and noun phrase; 87A: Order of adjective and noun; 93A: position of interrogative phrases in content question. Note: The results for 85A are based on manual evaluation as the top silhouette scores failed to give the correct clusters. Closeness C = Closeness Centrality; Neighbourhood C = Neighbourhood connectivity.

4 Discussion and Conclusion

Our work provides some support that word order generalisations are encoded in a network and can be automatically derived from it. In particular, the results from experiment 1 showed that when the Subject-Object-Verb orders found in the Greenbergian universals are probed through the combinatorial nodes, the correct word order typologies could be found. In addition, experiment 2 showed that similar (combinatorial) node-parameters lead to the right language clusters. We found that simply by inducing verb order and using the appropriate parameters, we can derive other linguistic order which share implicational relations with the verb order. These results are in accordance with the claim that networks are a meaningful representation of a linguistic knowledge. The nodes which led to the best classification based on a particular feature were major word orders, e.g., SOV, SVO, SVIO, etc. It is interesting to notice that including the order of indirect object induced certain linguistic features in Layer 2.

Our analysis was affected by multiple factors such as the treebank size, alignment of languages in UD and WALS, etc. For example, the silhouette score is higher when clusters are dense and well-separated. Since the cluster sizes are non-uniform, so is the density of clusters which is a function of the number of points in a cluster. The number of points in a cluster follows a power law, which is the primary reason for the non-uniformity in the cluster sizes. We also saw that the analysis in experiment 1 failed to induce any VSO-order based universal since no language considered has a dominant VSO order in the respective treebank. Similarly, while the ‘Outperc’ parameter that encodes the combinatorial property of the nodes in layer 2 was quite effective in classifying languages, in some cases where there is no dominant verb order pattern, ‘Out-Degree’ helps. While both ‘Outperc’ and ‘Out-Degree’ are very easy to interpret, other parameters such as ‘Eccentricity’, ‘NeighborhoodConnectivity’, that also lead to good clusters, are less transparent in their interpretability vis-à-vis linguistic generalizations. Indeed, the fact that the language network in this work lends itself to interpretability is a very attractive feature of this approach. Since, the network’s properties and the representation is tractable, we can investigate the linguistic validity of various parameters. While the current work has shown some promise wrt capturing simple word order generalizations, it remains to be seen if such a representation can capture other complex linguistic constraints.

References

- Olga Abramov and Alexander Mehler. 2011. Automatic language classification by means of syntactic dependency networks. *Journal of Quantitative Linguistics*, 18(4):291–336.
- Željko Agić, Maria Jesus Aranzabe, Aitziber Atutxa, Cristina Bosco, Jinho Choi, Marie-Catherine de Marnette, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Jan Hajič, Anders Trærup Johannsen, Jenna Kanerva, Juha Kuokkala, Veronika Laippala, Alessandro Lenci, Krister Lindén, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Héctor Alonso Martínez, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Joakim Nivre, Hanna Nurmi, Petya Osenova, Slav Petrov, Jussi Piitulainen, Barbara Plank, Prokopis Prokopidis, Sampo Pyysalo, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Kiril Simov, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. 2015. Universal dependencies 1.1. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.
- Réka Albert, Hawoong Jeong, and Albert-László Barabási. 2000. Error and attack tolerance of complex networks. *nature*, 406(6794):378.
- Paolo Allegrini, Paolo Grigolini, and Luigi Palatella. 2004. Intermittency and scale-free networks: a dynamical model for human language complexity. *Chaos, Solitons & Fractals*, 20(1):95–105.
- Albert-László Barabási. 2011. The network takeover. *Nature Physics*, 8(1):14.
- Javier Borge-Holthoefer and Alex Arenas. 2010. Semantic networks: Structure and dynamics. *Entropy*, 12(5):1264–1302.
- Guido Caldarelli. 2007. *Scale-free networks: complex webs in nature and technology*. Oxford University Press.
- Ramon Ferrer-i Cancho, Andrea Capocci, and Guido Caldarelli. 2007. Spectral methods cluster words of the same class in a syntactic dependency network. *International Journal of Bifurcation and Chaos*, 17(07):2453–2463.
- Ramon Ferrer-i Cancho, Ricard V. Solé, and Reinhard Köhler. 2004. Patterns in syntactic dependency networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 69 5 Pt 1:051915.
- Ramon Ferrer-i Cancho and Richard V Solé. 2001. The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1482):2261–2265.
- Monojit Choudhury, Diptesh Chatterjee, and Animesh Mukherjee. 2010. Global topology of word co-occurrence networks: Beyond the two-regime power-law. In *Proceedings of the 23rd international conference on computational linguistics: Posters*, pages 162–170. Association for Computational Linguistics.
- Cindy Chung and James W Pennebaker. 2007. The psychological functions of function words. *Social communication*, 1:343–359.
- Allan M. Collins and Elizabeth Loftus. 1975. A spreading activation theory of semantic processing. *Psychological Review*, 82:407–428.
- Jin Cong and Haitao Liu. 2014. Approaching human language with complex networks. *Physics of Life Reviews*, 11(4):598 – 618.
- Luciano da Fontoura Costa, Osvaldo N Oliveira Jr, Gonzalo Travieso, Francisco Aparecido Rodrigues, Paulino Ribeiro Villas Boas, Lucas Antikeira, Matheus Palhares Viana, and Luis Enrique Correa Rocha. 2011. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. *Advances in Physics*, 60(3):329–412.

- Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.
- Joseph H Greenberg. 1963. Some universals of grammar with particular reference to the order of meaningful elements. *Universals of language*, 2:73–113.
- Jinyun Ke and YAO Yao. 2008. Analysing language development from a network approach. *Journal of Quantitative Linguistics*, 15(1):70–99.
- Alan J Lerner, Paula K Ogrocki, and Peter J Thomas. 2009. Network graph analysis of category fluency testing. *Cognitive and Behavioral Neurology*, 22(1):45–52.
- Haitao Liu and Wenwen Li. 2010. Language clusters based on linguistic complex networks. *Chinese Science Bulletin*, 55(30):3458–3465.
- Haitao Liu and Chunshan Xu. 2011. Can syntactic networks indicate morphological complexity of a language? *EPL (Europhysics Letters)*, 93:28005.
- Ken McRae and Stephen Boisvert. 1998. Automatic semantic similarity priming. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24:558–572.
- Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris Mcnorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547–559.
- Alexander Mehler, Andy Lüicking, Sven Banisch, Philippe Blanchard, and Barbara Job, editors. 2016. *Towards a Theoretical Framework for Analyzing Complex Linguistic Networks*. Springer.
- Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.
- George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Glyn Morrill. 2000. Incremental processing and acceptability. *Computational linguistics*, 26(3):319–338.
- Mark Newman. 2010. *Networks: an introduction*. Oxford university press.
- 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*.
- Beth A. Ober and Greg Shenaut. 2006. Semantic memory. *Handbook of Psycholinguistics*, pages 403–453.
- Romualdo Pastor-Satorras and Alessandro Vespignani. 2007. *Evolution and structure of the Internet: A statistical physics approach*. Cambridge University Press.
- M. J. Pickering and V. S. Ferreira. 2008. Structural priming: A critical review. *Psychological Bulletin*, 134(3):427–459.
- Martin J. Pickering and Holly P. Branigan. 1998. The representation of verbs: Evidence from syntactic priming in language production. *Journal of Memory and Language*, 39:633–651.
- Ardi Roelofs. 1992. A spreading-activation theory of lemma retrieval in speaking. *Cognition*, 42(1-3):107–142.
- Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

- P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, Schwikowski B., and T. Ideker. 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504.
- Cynthia S Q Siew, Dirk U Wulff, Nicole Beckage, and Yoed Kenett. 2018. Cognitive network science: A review of research on cognition through the lens of network representations, processes, and dynamics.
- Ricard V Solé, Bernat Corominas-Murtra, Sergi Valverde, and Luc Steels. 2010. Language networks: Their structure, function, and evolution. *Complexity*, 15(6):20–26.
- K. M. Tooley and M. J. Traxler. 2010. Syntactic priming effects in comprehension: A critical review. *Language and Linguistics Compass*, 4(10):925–937.
- Michael S Vitevitch. 2008. What can graph theory tell us about word learning and lexical retrieval? *Journal of Speech, Language, and Hearing Research*.
- Michael S Vitevitch, Gunes Ercal, and Bhargav Adagarla. 2011. Simulating retrieval from a highly clustered network: Implications for spoken word recognition. *Frontiers in psychology*, 2:369.
- Radek Čech, Ján Mačutek, and Zdeněk Žabokrtský. 2011. The role of syntax in complex networks: Local and global importance of verbs in a syntactic dependency network. *Physica A: Statistical Mechanics and its Applications*, 390(20):3614 – 3623.

Parallel Dependency Treebank Annotated with Interlinked Verbal Synonym Classes and Roles

Zdeňka Urešová

Eva Fučíková

Eva Hajičová

Jan Hajič

Charles University

Faculty of Mathematics and Physics

Institute of Formal and Applied Linguistics

Malostranske nam. 25

11800 Prague, Czech Republic

{uresova, fucikova, hajicova, hajic}@ufal.mff.cuni.cz

Abstract

We present an ongoing project of enriching an annotation of a parallel dependency treebank, namely the Prague Czech-English Dependency Treebank, with verb-centered semantic annotation using a bilingual synonym verb class lexicon, CzEngClass. This lexicon, in turn, links the predicate occurrences in the corpus to various external lexicons, such as FrameNet, VerbNet, PropBank frame files, OntoNotes, and WordNet. We briefly describe the content of the CzEngClass synonym class lexicon and then we focus on its use for an enrichment of corpus annotation, which proceeds in two steps - automatic preprocessing and manual correction. This paper describes a first milestone of a long-term project; so far, approx. 100 CzEngClass classes, containing about 1800 different verbs each for both Czech and English, are available for such annotation. The corpus coverage at the moment is about 50%, allowing us to extract some basic statistics and discover a set of issues that appeared during the annotation process. The ultimate goal is to have a high-coverage, multilingual verbal synonym lexicon and corpora with all events annotated by such lexicon, to serve both theoretical studies in lexical semantic, translatology, corpus annotation studies etc. as well as a usable resource for training automatic semantic text processing systems for event/participant detection and linking and for general information extraction.

1 Introduction

While there are various richly annotated corpora linked to lexicons for several languages, such as OntoNotes (Pradhan et al., 2007) or the Prague Dependency Treebank projects (Hajič et al., 2006; Hajič et al., 2018), there are only a few that link substantial amount of annotated material to semantic lexicons, such as FrameNet, VerbNet, SemLink, PropBank or WordNet, and to our knowledge none that would link to all of them within a single corpus.

The project presented in this paper aims at filling this gap. The aim is to create a richly annotated corpus where each occurrence of a verb, for example *say*, is annotated by a (bi-lingual) synonym class *say, tell, disclose, report, ..., říci, sdělit, uvést, ...* and its dependents in the semantic representation (regardless of their syntactic realization) are labeled by semantic roles assigned to that class (Speaker, Addressee, Information).

Such a resource can be divided into two components:

- a semantically oriented bi- or multilingual verbal synonym lexicon, linked to all the other lexical resources, and
- the richly annotated corpus that contains references to entries in this semantic lexicon at every content verb (predicate) in the corpus.

The first component is covered by the existing CzEngClass lexicon¹ (Urešová et al., 2018c) which, while not complete and covering only Czech and English at this time, already provides enough synonym classes (and promises more coverage in the future) to approach the annotation task (the 2nd point above).

In this paper, we start with a short description of the resources used directly or indirectly through the available lexicon and corpus (Sect. 2), then we show how we have proceeded with the annotation process

¹<http://hdl.handle.net/11234/1-2977>.

(Sect. 3) and present the basic statistics of the automatic part of the annotation part of the process as applied to the whole corpus (Sect. 4). Manual corrections performed on a sample of 100 verb-pair occurrences are described in Sect. 5, giving a first glimpse of the effort needed to complete the manual part of the annotation for the whole corpus. Sect. 6 describes some lessons learned, summarizes the findings and provides some hints for future work.

2 Original Resources Used

The main resource is the CzEngClass lexicon. Its entries serve as the target of reference links attached to verb occurrences in the corpus (Urešová et al., 2018a).

2.1 The Lexicons

2.1.1 The CzEngClass Lexicon

The CzEngClass lexicon has the following structure (Fig. 1):

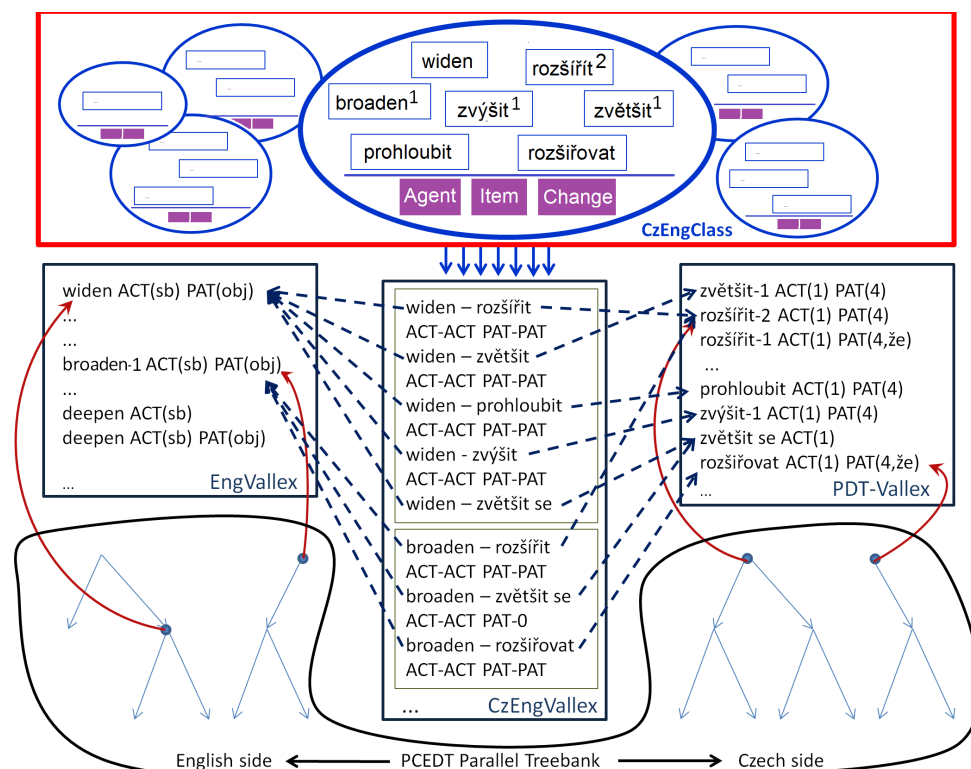


Figure 1: CzEngClass lexicon & other resources (from (Urešová et al., 2018a))

The lexicon consists of cross-lingual synonym classes which group verb senses (of different verbs) that have the same or similar meaning² and the (valency) arguments of which can be mapped to a common set of semantic roles, called a *Roleset*. The semantic roles (SRs) are assigned to all the members of one synonym class, and mapped individually to their valency arguments as captured for those verb senses in the EngVallex (Cinková et al., 2014) and PDT-Vallex (Hajič et al., 2003) lexicons.³ In Fig. 1, the lexicons are depicted as the square boxes on the left and right, just below the CzEngClass core lexicon depiction on top in the big rectangular box, where each oval shows one synonym class; the purple rectangular boxes on the bottom of one class present the common set of semantic roles for that class: Agent, Item, Change).

²The notion of “same or similar meaning” is used here rather intuitively, as the understanding of “synonymy” itself varies quite substantially. However, adding the mapping condition between roles and arguments helps to use more substantiated and evidenced criterion for deciding which verbs belong to the particular class.

³The valency lexicons use labels like ACT for Actor – the first argument, and PAT for Patient – the second argument, (Sgall et al., 1986).

The SRs are simultaneously linked to the existing verbal pairings (translational equivalents) found in the CzEngVallex lexicon (Urešová et al., 2016). CzEngVallex (the large box in the middle of Fig. 1) is in turn linked to the PCEDT parallel corpus (see the bottom of Fig. 1). In addition, CzEngClass entries refer also to several existing semantic lexicons (Sect. 2.1.2). More details on the mapping of SRs to valency slots of the corresponding valency lexicon entries are presented later in this paper, with examples in Tables 1 and 2.

The examples show some of the basic properties of the entries in the CzEngClass lexicon, and illustrate some specific issues that had to be dealt with.

Class: <i>surprise – překvapit</i>		
Class Member (sense ID)	Roleset (semantic roles)	
	Experiencer	Stimulus
surprise (EngVallex-ID-ev-w3269f1)	PAT	ACT and/or MEANS
překvapit (PDT-Vallex-ID-v-w4862f1)	PAT	ACT and/or MEANS
ohromit (PDT-Vallex-ID-v-w3015f1)	PAT	ACT and/or MEANS

Table 1: Example verbal synonym class for *surprise – překvapit* with role mappings (simplified)

In Table 1, a relatively “regular” (and small) synonym class is presented. This class (*surprise – překvapit*) bears two semantic roles: Experiencer and Stimulus. These roles are mapped, for each class member, to the valency slots associated with the individual verbs in the PDT-Vallex and EngVallex valency lexicons. In this class, the mapping is the same for all verbs in the class. We can also observe here a non-trivial (non-1:1) mapping, namely that the Stimulus is not always expressed by an ACTor alone. For example, in *Mr. X. ACT surprised Mr. Y. PAT by claiming. MEANS the prize for himself.*, the role Stimulus is formed by joining of both Mr. X and the “claiming event” (for which Mr. X is in fact the ACTor).⁴

In Table 2, the class *decline – odmítnout* exposes another frequent phenomenon, namely that the same role is mapped to different valency slots with different class members (ADDR vs. ORIG; in other classes, frequent pairs mapped to the same role are DIR1 and ORIG or ACT and LOC). This is mostly caused by the principles and conventions used in the underlying FGD valency theory (Sgall et al., 1986; Panevová, 1974), as reflected in the valency lexicons. At the same time, it exemplifies that *from the semantic perspective*, the valency slot labeling as determined by the rules and conventions of the FGD theory (or any other valency theory, for that matter) is not crucial, since the mapping provides the flexibility to relate them to the right semantic role(s). This example also shows that some verbs are included in that class, even if they do not express some semantic role from the Roleset by using a clearly assigned valency slot or any other modifier. Such a role (here, the Proposer) is deemed to be necessarily understood from a wider context.⁵ For example, *deny* (more precisely, its sense identified by EngVallex-ID-ev-ev-w876f1) and *refuse* (EngVallex-ID-ev-w@2598f1) both being without an obligatory ADDR, display this behavior. The (semantic) complementation assigned to the role of Proposer is for such cases marked as #sb(“somebody”), and it is assumed that in the process of corpus annotation, it will be inserted to the resulting representation and connected by a (semantic) co-reference link to the actual Proposer. The last column (Restrictions or requirements) of the Table 2 may contain additional requirements on the class member or its mapping to be valid, such as negation (*přijmout* in Czech means *accept*, cf. last line of Table 2).

2.1.2 External Lexicons

The CzEngClass lexicon, as mentioned above, indirectly refers each verb at each entry to the following external lexicons (Urešová et al., 2018a):

- The Berkeley FrameNet (Baker et al., 1998; Ruppenhofer et al., 2006; Fillmore, 1976; Fillmore, 1977), a lexical database of English,⁶

⁴This example does not claim anything special about (non-)agentive subjects - it rather shows that mapping between SRs and valency slots does not have to be necessarily 1:1.

⁵For such cases, CzEngClass uses specific pseudo-functors, such as #any,#sb, and #sth.

⁶<https://framenet.icsi.berkeley.edu>

Class: *decline – odmítnout*

Class Member (sense ID)	Roleset (semantic roles)			Restrictions or requirements
	Authority	Proposer	Proposal	
decline (EngVallex-ID-ev-w829f1)	ACT	ADDR	PAT	
deny (EngVallex-ID-ev-ev-w876f1)	ACT	#sb	PAT	
deny (EngVallex-ID-ev-ev-w876f2)	ACT	ADDR	PAT	
odmítnout (PDT-Vallex-ID-v-w2785f1)	ACT	#sb	PAT	
refuse (EngVallex-ID-ev-w@2598f1)	ACT	#sb	PAT	
přijmout (PDT-Vallex-ID-v-w5161f3)	ACT	ORIG	PAT	negation

Table 2: Example class for *decline – odmítnout* with role mappings (simplified, from 24 verbs)

- VerbNet (Schuler, 2006; Duffield et al., 2007; Kipper et al., 2006), a class-based verb lexicon⁷ with syntactic and semantic information of English verbs,
- PropBank (Palmer et al., 2005), linked to the OntoNotes corpus,⁸
- SemLink (Palmer, 2009)⁹ which connects the above lexicons, and
- WordNet (Miller, 1995; Fellbaum, 1998).¹⁰

The external links allow to compare and use these lexical resources with the annotated corpus, but for the proper semantic annotation are not used except as a source of secondary knowledge for the annotator when making annotation decisions.¹¹

2.2 The PCEDT Corpus

For this project, the Prague Czech-English Dependency Treebank (PCEDT), as available from LDC¹², described in (Hajič et al., 2012), is used. It contains about 55,000 sentences on each language side. The English side is the Wall Street Journal part of the Penn Treebank and the Czech side is its translation. Both sides are annotated for the so-called *Tectogrammatical Representation* (TR), used for the Prague Dependency Treebank family of projects (Hajič et al., 2006). Most importantly, content verbs are annotated by their corresponding valency lexicon entries as captured in the PDT-Vallex lexicon (Urešová et al., 2014) on the Czech side and in the EngVallex (Cinková et al., 2014; Cinková, 2006) on the English side.

As described in (Urešová et al., 2018a) (and in Sect. 2.1 above), the PCEDT corpus has been used as a source information for building the CzEngClass lexicon, raising the question of why the annotation cannot be fully deterministic. However, the classes are in principle independent of the PCEDT data and they underwent manual pruning; it is thus likely we get ambiguous (or even no) annotation by simply following the links from the CzEngClass entries directly to the corpus. In any case, the coverage of the CzEngClass entries will be relatively high, since they have been extracted from the same corpus in the first place.

3 The Annotation Process

3.1 Data Structure for Added Node Attributes (Technical Description, for Reference)

For the annotation process, we have used the valency reference IDs of individual verbs captured in both appropriate valency lexicons and also in the CzEngClass classes. In the PCEDT corpus, both on the Czech as well as English side, each occurrence of a content verb is annotated with such a valency frame ID. It is thus straightforward to “inverse” the mapping automatically, and include a reference to the class ID with

⁷<http://verbs.colorado.edu/~mpalmer/projects/verbnet.html>

⁸<http://propbank.github.io/>

⁹<https://verbs.colorado.edu/semlink/>

¹⁰<https://wordnet.princeton.edu/>

¹¹It should be noted that the coverage of these lexicons, at least based on the links found in CzEngClass, is not sufficient to allow for a systematic use in the annotation process, both automatic and manual.

¹²<https://catalog.ldc.upenn.edu/LDC2012T08>

each content verb (or at least to those contained already in CzEngClass). The “inversion” is meant with reference to the Fig. 1, where the arrows are driven top down. In the present project, the goal is to have direct reference links *from* the corpus at the bottom of Fig. 1 to the individual entries in the red box on the top. The CzEngClass resource in its current version 0.3¹³ assigns IDs to each verb in every class; these IDs are being used as the final reference of the verbs in the PCEDT. The annotation schema (at the highest, TR level) has been extended by the items (attributes) listed in Table 3.

Attribute	Description
<code>syn_class</code>	root attribute container for the semantic reference
<code>syn_class/class</code>	class container (of current node)
<code>.../class/class.rf</code>	ID of the class
<code>.../class/rep</code>	human-readable class name(s)
<code>syn_class/semrel</code>	sem. role container (rel. to parent node)
<code>.../semrel/semrole</code>	semantic role
<code>.../semrel/form</code>	required form
<code>.../semrel/spec</code>	additional information
<code>.../semrel/fromclass.rf</code>	class to which the role belongs

Table 3: The attributes for semantic (synonym classes) extension of Tectogrammatical Representation

The `syn_class` structured attribute contains either the class reference (`class.rf`), or the appropriate semantic role (`semrole`), or both (for predicates that are at the same time arguments to other predicates, typically roots of embedded clauses). The `fromclass.rf` attribute of each semantic role is necessary for temporarily ambiguous assignment of classes to its effective parent predicate, or in case of multiple effective parents (in coordination structures etc.), for keeping the same distinction permanent in the annotation.

3.2 Assignment of Classes

The initial assignment of CzEngClass classes to the verb occurrences in the corpus has been done automatically. First, for each occurrence of a content verb in the corpus, its valency reference is retrieved and searched for in the CzEngClass lexicon (checking each member in each class). If found, the class to which this class member belongs is recorded with the content verb node in the corpus (the `syn_class/class/class.rf` attribute) and the other attribute (`rep`) is filled appropriately.

As already explained in part in the last paragraph of Sect. 2.2, it is possible that a verb (as identified by its valency frame ID) is found in more than one synonym class, and therefore that each occurrence of that verb is annotated by several classes. In theory, this should not happen if the valency lexicon entries distinguish all possible verb senses, as they in principle should (Hajič et al., 2003). However, as the “senses” are defined, within the valency theory used, at the linguistic meaning level (and not fully semantically), there is no contradiction in the fact that some valency entries appear in several (semantically defined) synonym classes. Thus, there is no reason to “blame” the valency lexicons in such a case, and semantic differences will have to - naturally - be resolved during (semantic) corpus annotation.

3.3 Assignment of Semantic Roles

After an appropriate CzEngClass class is assigned to a verb node in the semantic representation of the corpus, the arguments are mapped to the semantic roles associated with that class and they are also filled into the `syn_class/semrel` attributes of the `syn_class` structure of the argument’s nodes. The automatic part of the assignment proceeds as follows:

- all the argument nodes of the given predicate are identified, based on the valency frame of the predicate verb as originally recorded in the corpus;

¹³<http://hdl.handle.net/11234/1-2977>

- the functor of every identified argument node is assigned the appropriate semantic role based on the CzEngClass lexicon mapping of semantic roles to arguments for the appropriate verb (class member) entry;
- the role, the form, additional information and ID of the class to which the semantic role belongs are stored in the `syn_class/semrel` attributes of the appropriate argument node.

This process might not, however, lead to all roles being represented in the corpus. For roles that are - for the given predicate (class member) - not mapped to any argument, it is necessary to introduce a new node in the semantic representation. This concerns roles mapped to pseudo-functors `#sb`, `#sth`, and `#any`. In those cases, this new node gets a special “lemma” `#SitRef` (“situational reference”), its (pseudo-)functor is copied from the CzEngClass mapping and its semantic role is filled into this node’s `syn_class` attributes. Similarly, for optional arguments or free modifications (adjuncts) listed in the CzEngClass entry mappings (for the given verb) which have not been found in the TR of the sentence in the corpus, a new artificial node is inserted. This node gets also the `#SitRef` “lemma”, the appropriate functor based on the CzEngClass mapping, and the corresponding semantic role; all filled into this new node’s `syn_class` attributes. This is similar to the approach to implicit semantic roles (“Null Instantiations”) described in (Ruppenhofer et al., 2009); the difference is that in our approach, licensing of such elements is based strictly on the set of roles assigned to the class (not on English - or any other language’s - grammar, given that we aim at multilingual classes). Also, we do not distinguish types of such situational references (indefinite, definite, ...) and defer this to the future process of discourse-based linking of the `#SitRef` to their referents, again without taking (grammatical) licensing into account; existence of a link will then correspond to definite null instantiations.

The `#SitRef` nodes are not created when two (or more) mappings exist for a given semantic role, and not every functor from these mappings is found in the corpus sentence representation. In such a case, only those present in the corpus are assigned a semantic role from the CzEngClass entry, and no new nodes are created.

However, if, for a given semantic role, no node in the corpus exists to which it is mapped in the CzEngClass lexicon, only one `#SitRef` node is created, namely the one corresponding to the functor with the highest precedence, where precedence is heuristically defined in the following way:

- core arguments in the order ACT, PAT, ADDR, EFF and ORIG;
- free modifications in the order BEN, SUBS, RCMP, MANN, LOC, TWHEN, DIR3, CAUS, AIM;
- all other free modifications in alphabetical order.

In the case of repeated free modifications in the corpus, only the first one (leftmost) is assigned the appropriate mapped semantic role.

4 Properties of the Enriched Corpus

After the automatic assignment of the “inverted” references, statistics have been collected. For certain configurations, examples have been extracted and an initial manual inspection performed.

4.1 Basic Statistics

About 50% of verbs annotated with the original valency lexicon entry in the corpus have received a CzEngClass ID (67,733 out of 130,079 on the English side and 48,445 out of 118,029 on the Czech side).

However, only 33,005 English (32,560 Czech) verbs are aligned with a Czech (English) verb found in CzEngClass.¹⁴ In conclusion, the coverage of the corpus by the current version of CzEngClass is about half of the corpus in terms of independent coverage of its Czech and English side, but only slightly above 25% when also the bilingual alignment is taken into account.

Up to five classes have been assigned to a single verb node in the corpus; i.e., there are verbs (verb senses) in both Czech and English that appear in (up to) five CzEngClass synonym classes. While the 1:1 alignment

¹⁴The asymmetry between the two last numbers is due to non-1:1 verb alignments.

prevails (in about one third of the cases where the aligned verbs are both found in CzEngClass), there are nontrivial numbers of occurrences of a 2:2, 1:2, 2:1, 3:2 etc. alignments.

Finally, of those 27,242 pairs aligned n:n (only 1:1, 2:2 and 3:3 alignments found), 21,050 CzEngClass class pairs fully matched between the two languages.

4.2 Manual Inspection and Examples of Mismatch

The fully matching pairs (i.e., those 21,050 matching pairs, or more precisely the 16,825 1:1-aligned full matches) are in fact what is to be expected, should the bilingual semantic synonym lexicon be “nice and clean.” However, not unexpectedly, language(s) do(es) not behave that nicely. It is therefore interesting to investigate the other cases.

Manual inspection of the non-1:1, non-matching cases revealed the following:

- for any non-1:1 alignment, i.e., for cases when the Czech or English verb or both are in more than one class: either the classes should be merged (as is often the case, e.g., for the verbs of communication, as we have acknowledged in (Urešová et al., 2018b) while describing an independent manual annotation experiment), or the verb sense distinctions as represented by the valency frames in the PDT-Vallex and EngVallex lexicons, as used for the PCEDT corpus annotation, are too coarse-grained and should be split into more verb senses;
- for an 1:1 alignment where the classes do not match, there are two possible causes:
 - the classes/alignments are plain wrong,
 - or the alignment is (sort of) OK, but the original sentence has been translated too freely, reformulating the source text to the extent that synonymy between the two “corresponding” verbs does not hold as defined in the CzEngClass specifications (Urešová et al., 2018a).¹⁵

For example, the aligned verb pair *say - uvést* has been (in many sentences) assigned three classes on the English side and two of them at the Czech side; closer inspection shows that these are to be merged.¹⁶

Example of a non-matching 1:1 class alignment is the pair *suggest - ukázat*: each side has been assigned a different class. Closer inspection shows that the Czech verb has semantically two different meanings - one is close to *suggest*, and the other one corresponds to *prove, implicate, establish, demonstrate, ...* which suggests that two senses of the Czech verb *ukázat* should be established (pun intended).

We are leaving out the cases where the original alignment does not strictly pair verbs (e.g., the translation is not literal, nominalization has been used, alignment error).

5 Manual Corrections

After the automatic part of the semantic annotation process has been completed, manual effort is needed to disambiguate and correct it.

One hundred paired verb occurrences in the parallel corpus have been selected from section 00 of the PCEDT (continuously to have wider context available).¹⁷ Only those aligned pairs that have been both automatically assigned at least one class have been considered. Out of these 100 pairs, i.e., 200 verb tokens (100 on the Czech side, 100 on the English side), 117 verb occurrences have been assigned multiple classes (up to 4 different ones), 48 in Czech and 69 in English. As a first step, these had to be manually disambiguated.

5.1 Removing Duplicate Classes

As it appears, many of the multiply assigned classes have in fact been the artifact of the CzEngClass lexicon construction - some classes are clearly duplicates and should have been merged. This concerns mostly the class *say - říci*, which is in fact assigned (to various verbs occurring in the corpus) in almost half the

¹⁵In fact, the translation could also be plain wrong, but we have not found such a case.

¹⁶It should be noted that the reason for having very similar classes that in fact should be just one class (i.e. to be merged in the process) is merely the way the classes have been created: each has been seeded by a randomly chosen verb, but some could have been synonyms, which could only be revealed later by the annotation process as described in (Urešová et al., 2018b) and here.

¹⁷Files *wsj0006* to *wsj0020*.

sentence pairs (46 out of the 100 pairs). The other class identified for such a merge in the lexicon is *require* - *vyžadovat*. After removing such duplicates, there remained 27 Czech and 60 English verb occurrences to manually disambiguate.

5.2 Manual Class Disambiguation

After the remaining 27 Czech and 60 English verbs have been disambiguated, statistics on the type of the disambiguation have been collected (Table 4).¹⁸

Disambiguation type:	More specific class selected	More general class selected	Competing selected
Czech	13	7	7
English	51	2	7
Total	64	9	14

Table 4: Statistics on manual class disambiguation results, by type

Closer inspection shows the following:

- most of the English cases where more specific meaning has been selected applies to the verb *say*, which appears both in the class *say* - *řici* as well as in the more general class *talk* - *mluvit*.¹⁹
- other examples where the more specific class has been selected are the classes *offer* - *nabídnout* and *provide* - *poskytnout*, which share, e.g., the verb *offer* itself, used in two different contexts: if the entity offered is a true offer which can be refused, it belongs to the more general class *offer* - *nabídnout*, whereas if the offer also means that it has to be 'accepted' unconditionally, then it has to be annotated by the class *provide* - *poskytnout*, as in: “[it] is the second incentive plan the magazine has offered advertisers in three years”²⁰, where the advertisers have no choice but to use this new plan (if they want to advertise in this particular magazine).
- examples of where the more general class of competing classes in a hierarchical relation is *pay* - *platit* vs. *repay* - *splatit*; these classes share a number of verbs, but depending on context, the more specific or the more general class must be selected. In the sentence “... to pay for the plant”²¹ the more general interpretation has been selected, since not even the textual context (discourse) makes it clear whether this payment is a “simple” payment, or a repayment of a loan or similar debt.²²
- an example of a selection among competing classes, where no hierarchy among them could be identified, are the following three classes, found three times in the annotated sample: *expect* - *čekat* (something from somebody), *anticipate* - *předpokládat* and *predict* - *předpovídat*, which share the verbs *expect*, *suppose*, *believe*, *očekávat*, *čekat* and others. While these verbs share valency frames across usages and interpretations, they are used in non-synonymous contexts - expecting (that someone does something which he/she should do or is planned to do) is not the same as predicting/forecasting (that something happens) and that is in turn not the same as anticipating (that something I believe happens actually happens). However, in this case no hierarchy could be determined among these classes.

¹⁸The purpose of this experiment was to find interesting cases, as opposed to measuring statistical variables such as inter-annotator agreement, for which there was not enough data. For larger but simpler annotation sample and its evaluation, see (Urešová et al., 2018b).

¹⁹There is no explicit hierarchy recorded in the CzEngClass lexicon yet, but the analysis of (not only) the ambiguous cases clearly shows that there must be one to be taken into account in the future.

²⁰File wsj0012, sentence 2

²¹File wsj0015, sentence 3

²²The actual full sentence reads: “In a disputed 1985 ruling, The Commerce Commission said Commonwealth Edison could raise its electricity rates by \$49 million to pay for the plant.”; we might speculate that from the fact that the company is acquiring the plant now while rates can only be raised in the future, it follows that it has to borrow money now and repay those \$49 million later, but that would be reaching too much into the world knowledge and even then, one cannot exclude that the conditions of the contract will be different and no borrowing in fact occurs. In such cases, the rule that we have adopted reads “use the more general class if no clear evidence is found to opt for the more specific one”.

The above points confirm, as already noted, that it will be practical to introduce a hierarchy into the (so far) flat list of classes in the CzEngClass lexicon. However, it seems that such a hierarchy will be slightly different than the one found e.g., in FrameNet, mainly because it will have no “container-only” classes (such as those found in FrameNet). The nodes in the “hierarchical tree” will be the classes themselves, and the tree will be used for guiding the annotation. We are leaving for future work to see if the evidence from the corpus annotation, as exemplified above, has any theoretical or methodological implications in the area of synonymy or lexical semantics in general.

5.3 Reassigning Semantic Roles

In some cases, the automatic assignment of semantic roles based on the CzEngClass lexicon failed; out of the 100 predicated verb occurrences examined (corresponding to 290 pairs of dependent aligned nodes holding semantic roles), 21 displayed a problem (7.2%). After an inspection, we determined a few types of failures:

- structural splitting of a semantic role:

This is a typical case for verbs of communication, where one semantic role can be expressed either as one valency argument (typically as PATient) or as two valency arguments (typically split into PATient and EFFect): *Paul said that he is.PAT-Information a liar.* vs. *Paul said about him.PAT-Information that he is.EFF-Information a liar.*

- multiple structural expression of a semantic role:

Some semantic roles can be syntactically expressed in multiple ways that are not mirrored in the valency frame. This is partially due to the valency theory used for the Tectogrammatical Representation. For example, the semantic role “Speaker” can be structurally expressed in multiple ways, such as ACTor or LOCcation: *He.ACT-Speaker called him a liar.* vs. *In The New York Times.LOC-Speaker, he was called a liar.*²³

- semantic roles reassigned to other nodes (not directly dependent on verb); for example, in a sentence containing “... *expect regulatory approval*”,²⁴ the semantic role Source for the verb *expect* is the *regulatory body*, but since it syntactically depends on *approval*, and not on the verb *expect* itself, the automatic assignment of roles based on the valency to role mappings could not identify it correctly.

5.4 Situational Reference

The newly introduced nodes identified by the “lemma” #SitRef are meant to be linked to the actual situational participant in the current sentence, or more likely somewhere else in the annotated document, much like textual co-reference is being marked in the tectogrammatical annotation of the corpus.²⁵ This situation appears not to be frequent, except for errors in the original annotation or in the underlying valency lexicons and for certain frequent verbs in classes with the Benefactive and Addressee roles. We are leaving this to the future work.²⁶

5.5 Example Annotation

Figs. 2 to 5 show two (occurrences of) English verbs in the corpus, *say* and *expect*, both before and after the manual annotation as described in Sect. 3 and 5, for the sentence “*The thrift holding company said it expects to obtain regulatory approval and complete the transaction by year-end.*” (File wsj0006, sentence 2).

In these Figures, the underlying section of the deep dependency tree as originally annotated using the tectogrammatical representation specification is shown in gray and node labels (lemma, functor) in black.

²³Although one can consider to use two different labels for one semantic role distinguishing the animacy here (Speaker vs. Medium), due to a similar phenomenon occurring at other verbs and synonymous classes the authors decided to keep only one label and in sentences like *NYT.ACT called him a liar*, the ACTor is still labeled as Speaker and not Medium.

²⁴File wsj0015, sentence 23

²⁵Unless they were reassigned in one of the previous steps of the manual revisions of the corpus.

²⁶For the purpose of the experiment described here, the #SitRef nodes have not been part of the evaluation.

Blue and brown arrows show textual and grammatical coreference links. The CO label suffix denotes parts of coordinated structure. The most relevant for the discussion here are the red and green node attributes: the classes assigned by the CzEngClass lexicon are in red, and the semantic roles are in green (please note that the semantic roles are complemented by the class identifier, also in green, to determine to which class this role belongs in case of multiple classes assigned to their parent verb node).

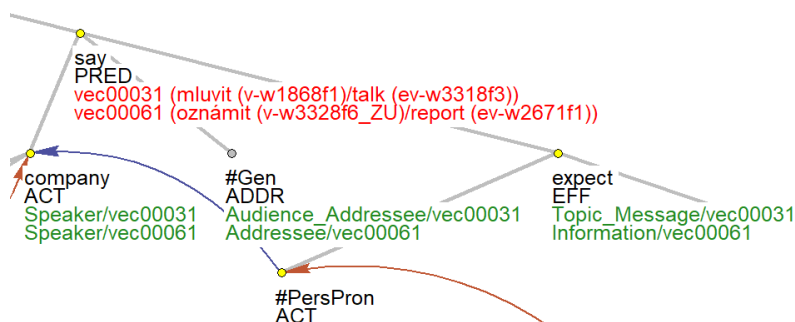


Figure 2: Automatic assignment of classes to *say* - ambiguity between classes 31 and 61

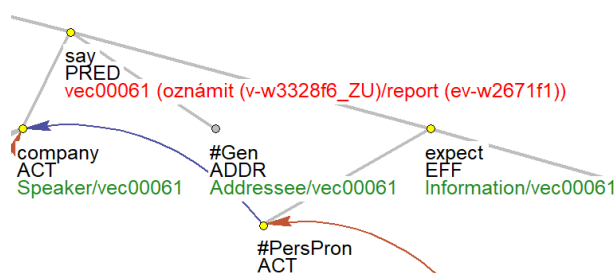


Figure 3: The verb *say* after disambiguation - class 61 and its roles selected

Figures 2 and 3 show a simple case where it was sufficient to disambiguate the appropriate class assigned automatically to the verb *say* in the example sentence, based on CzEngClass lexicon. Class 31 (*talk*) does not fit since the embedded clause is not just a topic, but a full information conveyed. Once disambiguated, the roles fit correctly.

Figures 4 and 5 show the more complex case of *expect*. First, there are three synonym classes to disambiguate: 2 (*expect*), 92 (*await*) and 93 (*assume*); the correct one in this case is class 2 (*expect*). For roles, it is necessary to link the newly generated #SitRef node to its (cognitive) antecedent, which is the *regulator*.

6 Conclusions and Future Work

We have described an experiment that enriched an existing annotated corpus by verb synonym classes, using a preliminary version of the CzEngClass lexicon. Even after the full lexicon is available, it is however expected that even the approx. half of the corpus (if the percentages from Sect. 4.1 can be extrapolated) which could have been automatically pre-annotated with the CzEngClass entries will need some manual inspection.

The verbs and their translations will, naturally, be never perfectly 1:1 aligned; we have shown some of the reasons and examples in Sect. 4.2. Assuming the identified duplicate classes are removed from the lexicon, there is about one-quarter of verb occurrences on the Czech side that are ambiguously annotated and need manual disambiguation; on the English side, this number was higher (60 out of the 100 sample occurrences), but this was due to a single verb - *say* - which might be perhaps tackled by introducing heuristics into the automatic pre-annotation procedure (e.g., select the class assigned on the Czech side if the aligned Czech verb is unambiguous and its class matches one of the English classes, possibly with additional restrictions).

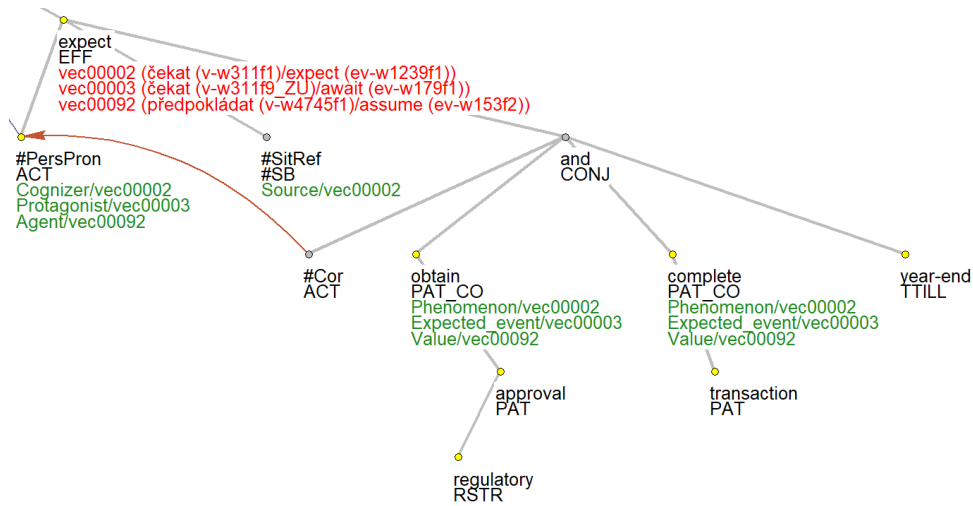


Figure 4: Automatic assignment of classes to *expect* - classes 2, 92 and 93

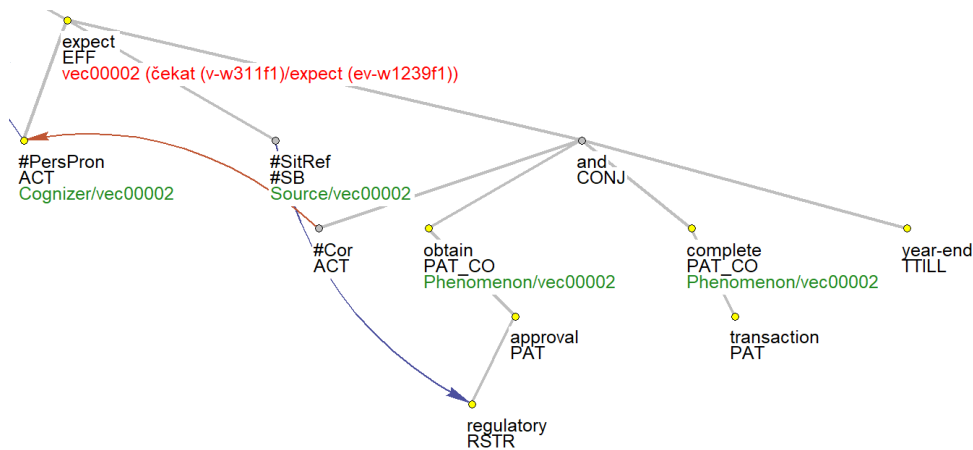


Figure 5: The verb *expect* after disambiguation - class 2 and its roles selected, #SitRef linked to *regulatory*

Similarly, the assignment of semantic roles to verb arguments and adjuncts as annotated in the corpus will need a manual pass. However, as the sample annotation has shown, the expected effort to correct errors in the semantic roles labeling part is relatively small - in the sample's 290 pre-assigned semantic roles, only 7.2% had to be corrected.

In the future, once the CzEngClass lexicon is published in full covering most, if not all, of the PCEDT corpus, the process described in Sect. 3 will be rerun, all the class alignments checked, and the resulting corpus will be published. This interlinked pair of resources will then be used for comparative lexical-semantic studies (also thanks to the links to the external lexicons, such as FrameNet, VerbNet, PropBank and WordNet), for study of translation from the lexical equivalence and synonymy perspective, and for machine learning experiments, e.g., for automatically extending the verb class synonym lexicon, and eventually for fully automatic annotation of (mono-, bi- and multilingual) corpora.

Acknowledgements

This work has been supported by the grant No. GA17-07313S of the Grant Agency of the Czech Republic. It uses resources hosted by the LINDAT/CLARIN (LINDAT/CLARIAH-CZ) Research Infrastructure, projects No. LM2015071 and LM2018101, supported by the Ministry of Education of the Czech Republic. Some of the resources have been also funded or co-funded by the European Commission through several projects of the 6th and the 7th Framework Programmes.

References

- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Silvie Cinková, Eva Fučíková, Jana Šindlerová, and Jan Hajič. 2014. *EngVallex - English Valency Lexicon*. LINDAT/CLARIN digital library. <http://hdl.handle.net/11858/00-097C-0000-0023-4337-2>.
- Silvie Cinková. 2006. From PropBank to EngValLex: adapting the PropBank-Lexicon to the valency theory of the functional generative description. In *Proceedings of LREC 2006, Genova, Italy*.
- Cecily Jill Duffield, Jena D. Hwang, Susan Windisch Brown, Dmitriy Dligach, Sarah E. Vieweg, Jenny Davis, and Martha Palmer. 2007. Criteria for the manual grouping of verb senses. In *Proceedings of the Linguistic Annotation Workshop, LAW '07*, pages 49–52, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA. 423 pp.
- Charles J. Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, 280(1):20–32.
- Charles J. Fillmore, 1977. *Scenes-and-frames semantics*, chapter 3, page 55 – 79. Number 59 in *Fundamental Studies in Computer Science*. North Holland Publishing.
- Jan Hajič, Jarmila Panevová, Zdeňka Urešová, Alevtina Bémová, Veronika Kolářová, and Petr Pajas. 2003. PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation. In Erhard Nivre, Joakim//Hinrichs, editor, *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, volume 9 of *Mathematical Modeling in Physics, Engineering and Cognitive Sciences*, pages 57–68, Vaxjo, Sweden. Vaxjo University Press.
- Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing prague czech-english dependency treebank 2.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3153–3160, Istanbul, Turkey. ELRA, European Language Resources Association.
- Jan Hajič, Eduard Bejček, Alevtina Bémová, Eva Buráňová, Eva Hajičová, Jiří Havelka, Petr Homola, Jiří Kárník, Václava Kettnerová, Natalia Klyueva, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Petr Pajas, Jarmila Panevová, Lucie Poláková, Magdaléna Rysová, Petr Sgall, Johanka Spoustová, Pavel Straňák, Pavlína Synková, Magda Ševčíková, Jan Štěpánek, Zdeňka Urešová, Barbora Vidová Hladká, Daniel Zeman, Šárka Zikánová, and Zdeněk Žabokrtský. 2018. *Prague Dependency Treebank 3.5*. Charles University, Prague, Czech Republic. LINDAT/CLARIN digital library. <http://hdl.handle.net/11234/1-2621>.
- Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, Magda Ševčíková Razímová, and Zdeňka Urešová. 2006. *Prague Dependency Treebank 2.0*. LDC, Philadelphia, PA, USA. <https://catalog.ldc.upenn.edu/LDC2006T01>, Catalog No. LDC2006T01.
- Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. Extending verbnet with novel verb classes. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 1027–1032, Genoa, Italy, May. European Language Resources Association (ELRA).
- George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, November.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106, March.
- Martha Palmer. 2009. SemLink: Linking PropBank, VerbNet and FrameNet. In *Proceedings of the Generative Lexicon Conference*, page 9 – 15.
- Jarmila Panevová. 1974. On verbal Frames in Functional Generative Description. *Prague Bulletin of Mathematical Linguistics*, 22:3–40.

- Sameer S. Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. Ontonotes: A unified relational semantic representation. *International Journal of Semantic Computing*, 01(04):405–419.
- Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. FrameNet II: Extended theory and practice. *Unpublished Manuscript*.
- Josef Ruppenhofer, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2009. Semeval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, SEW '09, pages 106–111, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Karin Kipper Schuler. 2006. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.
- Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The meaning of the sentence in its semantic and pragmatic aspects*. D. Reidel, Dordrecht.
- Zdeňka Urešová, Jan Štěpánek, Jan Hajič, Jarmila Panevová, and Marie Mikulová. 2014. *PDT-Vallex*. LINDAT/CLARIN digital library. <http://hdl.handle.net/11858/00-097C-0000-0023-4338-F>.
- Zdeňka Urešová, Eva Fučíková, and Jana Šindlerová. 2016. CzEngVallex: a bilingual Czech-English valency lexicon. *The Prague Bulletin of Mathematical Linguistics*, 105:17–50.
- Zdeňka Urešová, Eva Fučíková, Eva Hajičová, and Jan Hajič. 2018a. Creating a Verb Synonym Lexicon Based on a Parallel Corpus. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC'18)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).
- Zdeňka Urešová, Eva Fučíková, Eva Hajičová, and Jan Hajič. 2018b. Defining verbal synonyms: between syntax and semantics. In Dag Haug, Stephan Oepen, Lilja Ovrelid, Marie Candito, and Jan Hajič, editors, *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018) (Pub. No. 155)*, pages 75–90, Linköping, Sweden. Universitetet i Oslo, Linköping University Electronic Press.
- Zdeňka Urešová, Eva Fučíková, Eva Hajičová, and Jan Hajič. 2018c. Synonymy in Bilingual Context: The CzEng-Class Lexicon. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2456–2469.

Ordering of Adverbials of Time and Place in Grammars and in an Annotated English–Czech Parallel Corpus

Eva Hajičová, Jiří Mírovský, Kateřina Rysová

Charles University

Faculty of Mathematics and Physics

Institute of Formal and Applied Linguistics

[hajicova|mirovsky|rysova]@ufal.mff.cuni.cz

Abstract

The data from a parallel annotated English–Czech corpus serve for testing the general issue of the variability of the mutual position of LOC and TWHEN in Czech and English (Sect. 4.1) and for the analysis of the relation between information structure and the given order in the two languages (Sect. 4.2). The most relevant and innovative results in the investigation, namely the cases where the position of TWHEN and LOC differs in Czech and English in that the same modification is placed in Topic in the sentence in one language and in Focus in the corresponding sentence in the other are presented in Sect. 4.3.

1 Motivation and Research Question

In the early days of a massive entrance of corpus linguistics on the linguistic scene, C. J. Fillmore, in an attempt to characterize his own research position, compares two kinds of linguists: an armchair linguist and a corpus linguist. Fillmore (1992, 35) says: “Armchair linguist sits in his armchair, with his eyes closed and his hands clasped behind his back, once in a while, opens his eyes and shouts: Wow, what a neat fact”, while “Corpus linguist: has all of the primary facts he needs in the form of a corpus of approximately one zillion running words and he sees his job as that of deriving secondary facts from his primary facts.” And he concludes: “... the two kinds of linguists need each other. Or better, that the two kinds of linguists, wherever possible, should exist in the same body”. As for himself, he claims to be “an armchair linguist who refuses to give up his old ways but who finds profit in being a consumer of some of the resources that corpus linguists have created”.

In the era (and in the context) of treebanking, one can consider an armchair linguist to be a theoretically minded linguist and a corpus to be an annotated corpus in the form of treebanks, and it is in this sense that we have formulated our particular research question. The phenomenon under investigation is the *relation of word order and information structure*, the particular cases are *temporal and local modifications* of predicates and the data come from a *parallel English–Czech annotated corpus* (treebank).

The task we have faced is complicated by two facts: first, the information on structure is a very complex phenomenon and different approaches to its treatment have been proposed in theoretical literature since the pioneering studies by Czech scholars in the first half of the last century followed by such prominent linguists and semanticists as M.A.K. Halliday, B.H. Partee, M. Rooth, M. Krifka, E.F. Prince, K. Lambrecht, M. Steedman, E. Vallduví and E. Engdahl, to name just a few, and, second, it is hard to assess this phenomenon, so that annotation of information structure is very tricky (cf. Cook and Bildhouer, 2011) and therefore has to be carefully checked.

2 State of the Art

Though English representative grammars do not provide a systematic and comprehensive information on a possible variability of word order in English (which is quite understandable due to the predominance of grammatical factor determining the English SVO word order), it is somehow taken for granted, esp. in teaching English as a second language, that the unmarked order is SVOMPT, that is to say that with adverbials placed after the Object, Manner precedes Place and Place precedes Time. This more or less practical instruction is also reflected in Quirk et al. (1985, esp. parts 8.22–8.23): “Concerning adjuncts of the same grammatical class, subject to the stylistic and realizational factors already mentioned, will have their sequence determined by semantics and will normally appear in the order: process – space – time” (p. 650) giving examples such as *He worked at home that day.* or *The plane arrived uneventfully at Honolulu by midnight.* The authors continue: “Thus within the same class of adjuncts, those concerned with time are seen to be rather peripheral and this explains the case with which they can be moved to I (= initial position, EH): *By midnight, the plane arrived uneventfully at Honolulu.*” In the part on the relative positions of adjuncts (Chapter 8.87, pp. 565 ff.) the authors specify the order as respect – process – space – time – contingency, with two restrictions influenced by the information focus and the form of realization.

Leech and Svartvik (1994) mention the issues relevant for our investigation only briefly in the part on the position of adverbials (pp. 226–231) saying (p. 226) that “the place of an adverbial depends partly on its structure (whether it is an adverb, a prepositional phrase or clause, etc.), partly on its meaning (whether it denotes time, place, manner, degree, etc.). End-focus and end-weight also play a part.” The above-mentioned SVOMPT rule obtains here the following form: “When more than one of the main classes of adverbials occur in end-position, the normal order is manner/means/instrument + place + time.” The authors also take into account the influence of the form and the overall structure of the sentence, e.g. the fact that some adverbials which normally have an end-position can be in the front-position to avoid too many adverbials at the end of the sentence: *The whole morning he was working on his speech in the office.*

As for Czech, the relative freedom of surface word order makes it necessary to look for other than grammatical factors as determinants of the linear ordering of words in the sentence, the information structure being one of the main. In Vol. 3 of the representative Czech grammar *Mluvnice češtiny* (1987, p. 602) a “basic word order” is postulated, which is considered to be semantically based, reflecting the degrees of the so-called communicative dynamism (CD) as defined by the Czech anglicist Jan Firbas.¹ This basic word order may be influenced by the grammatical structure of the sentence, by its rhythmical structure and, marginally, by the size of the sentence elements in question. In the theory of information structure we subscribe to (the so-called topic–focus articulation, TFA, see e.g. Sgall et al., 1973; 1980; 1986) two orderings are postulated: one reflected in the surface shape of the sentence (surface word order) and the so-called underlying (deep) word order in the underlying (tectogrammatical) structure of the sentence. The underlying word order is semantically determined (and relevant), it reflects the TFA of the sentence and its counterpart in the surface is influenced, in addition to the TFA factors, by prosody, the overall structure of the sentence (e.g. the complexity of the structure), etc. One of the important notions introduced is the so-called systemic ordering (SO) as the order of verb modifications in the Focus part (see e.g. Sgall et al., 1980). The hypothesized order of main verb modifications is as follows: Actor – Temp – Cause – Regard – Aim – Manner – Accompaniment – Locative – Means – Addressee – Patient – Effect. The notion of SO in Focus is supposed to be universal, but the concrete order of modifications

¹ Cf. Firbas (1992, p. 105) “... the degree of CD carried by a linguistic element is the relative informational (communicative) value the element acquires in the development of the communication.”

may differ from language to language and has been already tested for some of them, see e.g. for German Sgall et al. (1995), for English Preinhaelterová (1997), for Czech Rysová (2014).

3 Methodology and Data

Our research question concerns the position of temporal and local modifications of predicates in Czech and English and the relation of this position to the information structure. The data come from a parallel English–Czech annotated corpus PCEDT (Hajič et al., 2012), which is a mostly manually annotated parallel corpus of English and Czech texts with almost 50 thousand sentences for each part. The E. part contains the Wall Street Journal section of the Penn Treebank (Marcus et al., 1993), along with the original phrase-structure analysis and a newly added dependency-based deep structure syntactic analysis (tectogrammatics). The Cz. part consists of manual translations of the original texts, along with their surface and deep syntactic analyses, automatically parsed and manually checked. We have analyzed the corpus findings and compared the results with claims made by existing representative grammars and other relevant studies and have tried to draw attention to contextual and other factors that play decisive role in the surface ordering of temporal (TWHEN) and locative (LOC) modifications. In doing so, we had in mind two limitations: the corpus data belong to the journalistic genre in which the TFA is not that clear as in other genres, and the translated sentences may be inclined to follow automatically the original order.

4 Queries and Corpus Findings

We have carried out a series of queries in which we were concerned with a general issue of variability of the mutual position of LOC and TWHEN in Czech and English (Sect. 4.1) and with the relation between TFA and the given order in the given languages (Sect. 4.2). The most relevant and innovative results in our investigation, namely the cases where the position of TWHEN and LOC differs in Czech and English in that the same modification is placed in Topic in the sentence in one language and in Focus in the corresponding sentence in the other are presented in Sect. 4.3.

4.1 Variability of the position of TWHEN and LOC

We have searched in the parallel corpus for cases with the Predicate as the root of the tree (excluding thus coordinated sentences) in which both TWHEN and LOC (occurring in the same tree) depend on the same Predicate. This search was carried out in the whole PCEDT, i.e. in the total of 39507 sentences with the Predicate as the root of the tree. The cases relevant for this step amount to 0.96% of the corpus. The results of our search are summarized in Table 1, where the E || Cz column refers to the number of cases in which the positions in Czech and English are the same.

It should be emphasized that the figures in Table 1 do not take into account the position of the modifications be it in the Topic or in the Focus, they just reflect the mutual positions of these modifications in the sentences in which both of them occur. The figures indicate that both orders are possible both in English and in Czech, and that in English the orders are relatively balanced (190 to 191), while in Czech the more frequent order is that of TWHEN before LOC (278 times when compared to LOC before TWHEN occurring 103 times).

	E Cz	E / Cz	Total E / Cz
LOC < TWHEN	85	105/18	190/103
TWHEN < LOC	173	18/105	191/278
Total	258	123	381

Table 1: The relative position of TWHEN and LOC in English and Czech in PCEDT

4.2 The relative position of TWHEN and LOC in the Focus part of the sentence

4.2.1 In the next step, we have taken into account the assumed division of the sentence into Topic and Focus and looked for cases in which both TWHEN and LOC were in the Focus part. The reasons why we have concentrated on the Focus part of the sentence, are twofold: first, and most importantly, we wanted to check whether and under which conditions the hypothesis of the above mentioned SO in the Focus is valid, both for English and for Czech, and, second, in this way, we could also check the before-mentioned general English word order “rule” SVOMPT, which indicates the order of Time after Place in the post-verbal position; with certain simplifications the post-verbal position may be considered to function as the Focus of the sentence.

We have tried first to search in that part of the PCEDT in which the sentences were annotated also as for their Topic–Focus articulation (3857 sentences), but the number of cases in which both TWHEN and LOC occurred in the same sentence in the relevant positions both in English and in Czech was very low (34 instances). Therefore we have decided to approximate the division into Topic and Focus as the position before (Topic) and after (Focus) the Predicate² and to carry out the search in the whole of PCEDT (on sentences with Predicate as the root of the tree), separately for English and for Czech.

	E.	E. after manual inspection	Cz.
TWHEN < LOC	129	103	164
LOC < TWHEN	202	130	90
TOTAL	331	233	254

Table 2: The occurrence of orderings of TWHEN and LOC in Focus in E. and in Cz. in PCEDT

The total number of sentences checked was 42717 for English and 39507 for Czech; the difference follows from the fact that there exist cases where one of the modifications is not realized by a separate sentence element.³ The results are given in Table 2.

The data obtained have made it possible to check the validity of the assumed so-called systemic ordering. The first attempt at such a verification for Czech was carried out by Rysová (2014) analyzing the data from the Prague Dependency Treebank 2.0 (PDT).⁴ The relevant figures in her Tables 6.1 (p. 77) and 6.10 (p.96) are summarized below in Table 3:

	Total number of occurrences	occurring in F	%
TWHEN	14552	4623	32
LOC	16948	10081	59
LOC < TWHEN		72	
TWHEN < LOC		332	

Table 3: The frequency of TWHEN and LOC (expressed by non-sentential elements) and their ordering in Focus in Czech in PDT 2.0 according to Rysová (2014)

² Such an approximation is based on the hypothesis common in many studies of information structure that the verb in principle stands on the boundary between the Topic and the Focus, cf. the notion of transition in Firbas (1992) and the analyses of Czech in Sgall et al. (1980) and Uhlířová (1974; 1987).

³ Comparing the English sentences containing LOC and TWHEN in PCEDT (without coordinated main predicates) with their Czech counterparts, LOC is missing in Cz. in 192 cases, and TWHEN is missing in Cz. in 88 cases. The difference is a consequence of several facts: the given modification in one language is translated by means of a different type of modification, a coordination structure is used in one language and not in the other, the given modification is understood as dependent on Noun rather than on a PRED, the dependency relations were understood differently, or a different structure is used in the translation.

⁴ The PDT (see the most recent version Hajič et al., 2018), contains approx. 50 thousand sentences of Czech journalistic texts annotated manually on several layers (morphology, surface and deep syntax) and contains also annotation of the topic-focus articulation of the sentences.

Rysová's results demonstrate that the data of PDT 2.0 support the SO as TWHEN < LOC; she also gives an explanation of the cases that do not correspond to this hypothesized order. Her observation is supported by the PCEDT data (see Table 2), though not so convincingly, which may be explained by the fact that the PCEDT data are translations and as such may mimicry to a considerable extent the E. order.

For English, our "raw" data indicate a different situation: TWHEN < LOC = 129 which is less than LOC < TWHEN = 202. However, after a manual inspection resulting in filtering out cases where the given modification, though placed after the verb, has to be characterized as contextually bound,⁵ i.e. belonging to the Topic part of the sentence, the figures attested were 103 for the TWHEN < LOC order, and 130 LOC < TWHEN order, which means that the preference for LOC < TWHEN is not so striking.

4.2.2 Let us first examine the examples of the TWHEN < LOC order, i.e. the order hypothesized by SO but counter to the assumed SVOMPT order. In 3 cases a decisive role was played by the form of the LOC modification as a clause (1).

- (1) *Researchers began using the drug in February. TWHEN on patients. LOC who had received kidney, liver, heart and pancreas transplants.*

In the remaining 100 cases the LOC modification can be supposed to exemplify the order as predicted by SO. In most of them, both modifications are short (or of a comparable length) so that the "weight" criterion cannot be applied, see (2).

- (2) *A volcano will erupt next month. TWHEN on the fabled Strip. LOC: a 60-foot mountain spewing smoke and flame every five minutes.*

With some examples, the TWHEN modification is closely related to the extralinguistic context (e.g. *today*) so that it can be understood as contextually bound and belonging to the Topic (3), though a different interpretation is also possible because in the preceding co-text District Court in Philadelphia is mentioned.

- (3) *The trial begins today. TWHEN in federal court. LOC in Philadelphia. LOC.*

4.2.3 As for the LOC < TWHEN order, i.e. the order counter to the SO but in concord with the assumed SVOMPT order, we have again put aside examples in which TWHEN was expressed by a clause, which certainly had an impact on its end-position. This group was much larger than in the previous case, namely there were 48 examples in which the TWHEN modification was expressed by a clause, see (4):

- (4) *Judy and I were in our back yard. LOC when the lawn started rolling like ocean waves. TWHEN*

The rest of the examples (82 sentences) mostly include the two modifications expressed by noun groups of a similar length (5), with an exception of some cases where the weight was a decisive factor (6).⁶

- (5) *Mr. Guber got his start in the movie business at Columbia. LOC two decades. TWHEN ago.*
(6) *WASHINGTON lies low. LOC after the stock market's roller-coaster ride. TWHEN.*

⁵ In the TFA theory, on which the TFA annotation is based (see e.g. Sgall et al., 1986), contextual boundness is a primary notion interpreted as follows: A contextually bound node represents an item presented by the speaker as referring to an entity assumed to be easily accessible by the hearer(s), i.e. more or less predictable, readily available to the hearers in their memory. Each element of the underlying dependency tree of a given sentence is assigned one of the values of the TFA attribute, namely cb (contextually bound non-contrastive), c (contextually bound contrastive) or nb (contextually non-bound).

⁶ As remarked by one of the reviewers, "lie low" may be understood rather as an idiomatic expression.

4.2.4 We have also made a random inspection for particular cases where the parallel English and Czech sentences differed in the ordering of the two modifications. Interestingly enough, there are cases for which we have not found any reason why this was so, except for the “different ordering principles” (7).

(7) E.: *The company was founded in Sacramento. LOC in 1929. TWHEN by two brothers, Ralph and Walter Merksamer, who operated as DeVon's Jewelers.*

Cz.: *Společnost založili v roce 1929. TWHEN v Sacramentu. LOC bratři Ralph a Walter Merksamerovi pod jménem DeVon's Jewelers.*

However, having in mind that our parallel corpus was composed of translations from English to Czech, there was no surprise that the “principle ordering” in the target language was not obeyed and the Czech translation copied the order in E., see (8):

(8) E.: *Mr. Guber got his start in the movie business at Columbia. LOC two decades ago. TWHEN.*

Cz.: *Guber začínal ve filmové branži v Columbia. LOC před dvěma desítkami let. TWHEN.*

To sum up, while the SO for Cz. has been supported by both the PDT and the PCEDT data, the data for E. provide a slight support for the SVOMPT order.

4.3 Differences between Czech and English in the placement of TWHEN or LOC in the Topic and in the Focus

Most interesting for our study are the cases, where the two languages studied differ in the placement of the modifications TWHEN or LOC in the Topic in one language and in the Focus part of the same sentence in the other. In order to get a richer sample of examples, we have searched in the whole of PCEDT and we have again approximated the division into Topic and Focus by the position of these modifications before (Topic) and after (Focus) the main verb (PRED). We have at our disposal the samples in Table 4.

	TWHEN	LOC
In E. before PRED, in Cz. after PRED	233	67
In E. after PRED, in Cz. before PRED	765	271
TOTAL different order in E. and Cz.	998	338

Table 4: The position of TWHEN and LOC with respect to the Predicate in English compared to Czech

4.3.1 The position of TWHEN

We have randomly chosen a sample of 100 E. sentences and their Cz. counterparts from each of the sets (out of 233 and 765 examples, respectively) and analyzed them, also with regard to the previous context. The following observations seem to hold:⁷

A. TWHEN after the Predicate in English and before the Predicate in Czech

(i) Typically, TWHEN is expressed in E. by a short adverb (*-ly* adverb, *yesterday*, ...) and is placed next to the Predicate. In such a case, this post-verbal element may be considered to be a part of Topic also in E.

(9) E.: *In national over-the-counter trading, the company closed yesterday at \$23.25 a share.*

Cz.: *Při celostátním mimoburzovním obchodování společnost včera uzavřela na 23.25.*

(ii) TWHEN is expressed in E. by a short adverb and placed at the end of the sentence, but (presumably) this adverb does not carry the intonation centre; these examples, if analyzed properly with regard to Top-

⁷ In the examples, the relevant elements are underlined.

ic and Focus rather than with regard to its pre- or post-verbal position, would not represent instances of differences we are looking for (10).

(10) E.: *Democrats had been negotiating with some Republican congressional leaders on a compromise lately.*

Cz.: *V poslední době vyjednávali demokraté s některými čelními republikánskými představiteli Kongresu o kompromisu.*

(iii) In E., the position of TWHEN at the end of the sentence (i.e. in the prototypical position of Focus) is due to the weight of the element, being a prepositional phrase or a whole dependent clause (11).

(11) E.: *The shares traded at about A\$ 1.50 in March, when the plan to acquire MGM/UA was announced.*

Cz.: *V březnu, kdy byl plán na převzetí společnosti MGM/UA oznámen, se akcie obchodovaly kolem 1,50 australského dolaru.*

(iv) Nevertheless there was a considerable number of “true” examples where the E. sentence differed from its Cz. equivalent in the placement of the TWHEN modification in the Topic vs. the Focus part (12), (13):

(12) E.: *Coke introduced a caffeine-free sugared cola based on its original formula in 1983.*

Cz.: *Coke v roce 1983 uvedla na trh bezkofeinovou slazenou kolu založenou na původní receptuře.*

(13) E.: *But losers were spread in a broad range by the end of the session.*

Cz.: *Ale koncem burzovního dne se rozšířily řady těch, co ztratili.*

For some of these cases, as (14), the initial position of TWHEN in Cz. may be interpreted as a contrastive Topic: it is still (a part of) Topic, the sentence being “about” it, but the contrastive character of this element makes it comparable with Focus (which, as a choice of alternatives, always has a contrastive character).

(14) E.: *But we're ... going to be in the exact same situation next year.*

Cz.: *Ale příští rok budeme... v naprosto stejné situaci.*

B. TWHEN before the Predicate in English and after the Predicate in Czech

(i) A tendency observed by Czech grammars was attested in our data, to place the Predicate into the second position of the Cz. sentence, which has led to the placement of the TWHEN modification after the verb also in case in which it was an indisputable element of the Topic of the sentence (15):

(15) E.: *A year earlier, Nationwide Health earned.PRED \$2.4 million, or 29 cents a share.*

Cz.: *Výnosy společnosti Nationwide Health činily.PRED v loňském roce 2.4 milionu dolarů, neboli 29 centů na akcii.*

(ii) The difference in the placement of the TWHEN modification is due to the preferred position of short adverbs in E. (16):

(16) E.: *The utility company currently has about 82.1 million shares outstanding.*

Cz.: *Tento podnik veřejných služeb má v současné době v oběhu 82.1 milionu akcií.*

(iii) However, even in this group, quite clear examples are found testifying the difference in Topic and Focus in E. and in Cz.; in some cases, the initial position should be understood as a contrastive Topic (17), see Quirk et al. (1985) where fronting is mentioned as a regular means for emphasizing a contrastive Topic.

(17) E.: *Only twice since the 1960s has annual gross domestic product growth here fallen below 5% for two or more consecutive years.*

Cz.: *Roční nárůst hrubého domácího produktu zde spadl pod 5 % během dvou nebo více po sobě jdoucích let pouze dvakrát od šedesátých let.*

4.3.2 The position of LOC

We have again randomly chosen 100 sentences from the set of LOC after PRED in E. and we have analyzed all the sentences in the set of LOC before PRED in E. (i.e. the total of 67 sentences) taking into consideration also the previous context. The following observations seem to hold:

A. LOC after the Predicate in English and before the Predicate in Czech

(i) As has been mentioned above in our discussion on the sentences with TWHEN, the position of a modification close to the Predicate may be considered as a part of Topic or alternatively as a part of Focus, as the example below demonstrates:

(18) E.: *The two boards said.PRED in a joint statement that the proposed merger agreement was considered in separate board meetings in Oslo Monday.*

Cz.: *Obě správní rady ve společném prohlášení uvedly.PRED, že navrhovaná dohoda o sloučení byla v pondělí posouzena na jednotlivých zasedáních správních rad v Oslu.*

(ii) The final position in E. need not be an indicator of the Focus position because the given element need not be a carrier of the intonation center; the prosodic factor is decisive here for the identification of Focus (19):

(19) E.: *Logic plays a minimal role here.*

Cz.: *Logika tady hraje minimální roli.*

(iii) A modification is placed at the end of the sentence in E. because of its weight, which does not necessarily mean that this modification is in Focus (20):

(20) E.: *The topic never comes up in ozone depletion "establishment" meetings, of which I have attended many.*

Cz.: *Toto téma se na „schvalovacích“ schůzích o ozónové díře, kterých jsem navštívil hodně, nikdy neujme.*

(iv) The placement of the modification is given by grammatical restrictions of word order in E., namely that subject should precede the verb (21); there belong also examples with *there*-construction (22):

(21) E.: *A tractor, his only mechanized equipment, stands in front of the pigsty.*

Cz.: *Před prasečím chlívem stojí traktor, jeho jediné mechanizované zařízení.*

(22) E.: *There was no new-issue activity in the derivative market.*

Cz.: *Na trhu odvozených cenných papírů nebyla vyvíjena žádná nová emisní aktivita.*

(v) The dependency relation is different in the original and in the translation: in E. “on television” depends on “events”, while in Cz. the LOC is understood as a modification of the verb (23).

(23) E.: *The Series typically is among the highest-rated sports events on television.*

Cz.: *V televizi světová série obvykle patří mezi nejvýše hodnocené sportovní události.*

(vi) However, similarly as is the case with the placement of the modification TWHEN, there was a considerable number of “true” examples where the original E. sentence differed from its Cz. equivalent in the placement of the LOC modification in the Topic vs. in the Focus part (24), (25).

(24) E.: *The citation was misstated in Friday's edition.*
Cz.: *V pátečním vydání byla tato citace uvedena chybně.*

(25) E.: *Each has an equal vote at the monthly meetings.*
Cz.: *Na měsíčních schůzích mají všichni stejný hlas.*

It is often the case that the preceding context helps to identify the Focus, but not necessarily so, as the following example demonstrates (26):

(26) E.: *The year was misstated in Friday's edition.*
Cz.: *V pátečním vydání byl rok uveden chybně.*

E. previous context: *QUANTUM CHEMICAL Corp.'s plant in Morris, Ill., is expected to resume production in early 1990.*

B. LOC before the Predicate in English and after the Predicate in Czech

The analysis of the examples with LOC before the Predicate in E. and after the predicate in Cz. has led to observations analogous to those mentioned in Sect 4.3.1 B above. Similarly as noted in 4.3.2 A, a tendency was also observed to place the Predicate into the second position of the Cz. Sentence, which has led to the post-verbal placement of the LOC modification also in case in which it was an indisputable element of the Topic of the sentence, see (27).

(27) E.: *In an interview, Pemberton Hutchinson, president and chief executive, cited several reasons for the improvement: higher employee productivity and “good natural conditions” in the mines, as well as lower costs for materials, administrative overhead and debt interest.*

Cz.: *Prezident a výkonný ředitel Pemberton Hutchinson jmenoval.PRED v rozhovoru několik důvodů zlepšení: vyšší produktivitu zaměstnanců a „dobré přírodní podmínky” v dolech, stejně jako nižší cenu materiálu, administrativní režii a úroky z úvěrů.*

Comparing the number of sentences in which the position of LOC with regard to the Topic and Focus position in Cz. and E. differed, it should be noted that in E., LOC occurred relatively much less frequently in the front position than in the Focus position (23% to 77%). Interestingly enough, almost the same proportion holds for TWHEN, which occurred in 20% in the front position and in 80% post-verbally. It seems that the final position of both of these modifications is the preferred one.

5 Summary

Our main concern has been the relation of word order and information structure in English and in Czech, in particular the mutual order of temporal and local modifications of predicates. We have put under scrutiny the data from the annotated parallel English–Czech treebank (PCEDT) and tested the variability of the order of the given types of modifications in general and two hypotheses on their preferential order in particular, namely the SVOMPT hypothesis for English and the so-called systemic ordering hypothesis for both languages. Our probe has demonstrated that corpus data offer much richer material to work with than an “arm-chair” linguist has ever had at her/his disposal but also that a careful manual check is necessary to obtain a reliable source for a detailed linguistic analysis that eventually may lead to some well-founded theoretical conclusions.

Acknowledgements

The authors are deeply indebted to Prof. Libuše Dušková, the leading Czech anglicist, for her observations and comments concerning the topic of this contribution.

The authors also gratefully acknowledge support from the Grant Agency of the Czech Republic (projects GA1703461S and GA17-06123S) and the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071). The research reported in the present contribution has been using language resources developed, stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (LM2015071 and OP VVV VI CZ.02.1.01/0.0/0.0/16 013/0001781).

References

- Cook, Philippa., Bildhauer Felix. (2011), Annotating Information Structure. The Case of “Topic”. In Dipper, S., Zinsmeister H. (eds.), *Beyond Semantics. Corpus-based Investigations of Pragmatic and Discourse Phenomena*. Bochum: Ruhr-Universität Bochum, 45–56.
- Fillmore, C. J. (1992), “Corpus linguistics” or “Computer-aided armchair linguistics”, In: *Directions in Corpus Linguistics*, Proceedings of Nobel Symposium 82 Stockholm, 4-8 August 1991, ed. Jan Svartvik, Mouton De Gruyter, Berlin New York, pp. 61–77.
- Firbas, Jan (1992), *Functional Sentence Perspective in Written and Spoken Communication*. Cambridge, Cambridge University Press.
- Hajič, Jan, Bejček Eduard, Bémová Alevtina et al. (2018), *Prague Dependency Treebank 3.5*. Data/Software. Institute of Formal and Applied Linguistics, LINDAT/CLARIN, Charles University. PID: <http://hdl.handle.net/11234/1-2621>.
- Hajič, Jan, Hajičová Eva, Panevová Jarmila et al. (2012), Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the LREC 2012*, pp. 3153–3160.
- Leech, Geoffrey and Jan Svartvik, 1994. *A Communicative Grammar of English*. 2nd ed., London: Longman.
- Marcus, M. P., Marcinkiewicz, M. A. and B. Santorini (1983), Building a Large Annotated Corpus of English. The Penn Treebank. *Computational Linguistics* 19 (2), pp. 313–330.
- Mluvnice češtiny* 3 (1987), Prague, Academia.
- Preinhaelterova, Ludmila (1997), Systemic ordering of complementations in English. *Philologica Pragensia* 1997, pp. 12–25.
- Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech and Jan Svartvik (1985), *A Comprehensive Grammar of the English Language*, London and New York: Longman.
- Rysová, Kateřina (2014), *O slovosledu z komunikačního pohledu* [On Word Order from a Communicative Point of View], Studies in Computational and Theoretical Linguistics 12, ÚFAL Charles University, Prague.
- Sgall, Petr, Hajičová, Eva and Eva Benešová (1973). *Topic, Focus, and Generative Semantics*, Kronberg/Taunus: Scriptor.
- Sgall, Petr, Hajičová, Eva and Eva Buráňová (1980), *Aktuální členění věty v češtině*. [Topic-Focus Articulation of the Czech Sentence]. Prague: Academia.
- Sgall, Petr, Hajičová, Eva and Jarmila Panevová (1986). *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Dordrecht: Reidel, Prague: Academia.
- Sgall, Petr, Oskar Pfeiffer, W. U. Dressler and Milan Půček (1995), Experimental research on systemic ordering, *Theoretical Linguistics* 21: pp. 197–239.
- Uhlířová, Ludmila (1974), O vztahu sémantiky příslovecného určení k aktuálnímu členění. [On the relation of semantics of adverbials to the information structure]. *Slovo a slovesnost* 35, pp. 99–106.
- Uhlířová, Ludmila (1987). *Knižka o slovosledu* [A book on word order]. Prague.

Weighted posets: Learning surface order from dependency trees

William Dyer

Oracle Corp

william.dyer@oracle.com

Abstract

This paper presents a novel algorithm for generating a surface word order for a sentence given its dependency tree using a two-stage process. Using dependency-based word embeddings and a Graph Neural Network, the algorithm first learns how to rewrite a dependency tree as a partially ordered set (poset) with edge-weights representing dependency distance. The subsequent topological sort of this poset reflects a surface word order. The algorithm is evaluated against a naive baseline of average dependency distances across 14 languages, performing well in terms of rank correlation and resulting rate of projectivity based on Universal Dependencies corpora.

1 Introduction

In a tradition dating at least back to Tesnière (1959), the words in a phrase or sentence can be thought of as a set of heads and dependents. Each word save the root is a dependent of another word, its head, and heads and dependents exist in a one-to-many relationship (Polguère and Mel’čuk, 2009). This arrangement of heads and their dependents forms a tree, or more formally an unordered directed acyclic graph (DAG), in which words are nodes and edges are the dependency relations. A sentence is one possible linearization or surface order of the DAG.

This paper describes a method for learning how to generate a valid¹ surface order from a dependency tree. Determining the underlying dependency tree from a surface order is the rather extensively studied task of parsing; this paper concerns the opposite task.

The key insight of the paper is that rather than learning to directly convert a dependency tree to surface order, the target is instead an edge-weighted partially ordered set (poset). The poset’s edge direction represents linear precedence in the surface order, while edge weight represents dependency distance, the number of words intervening between dependent and head in the surface order. The topological sort or linear extension of this poset—performed such that nodes connected by edges with smaller weights are placed closest to each other—reflects the surface order of the dependency tree.

For example, Figure 1 shows (a) the dependency tree, (b) edge-weighted poset, and (c) surface order of the sentence *Personally I recommend you take your money elsewhere*. Rather than attempting to learn how to convert (a) directly into (c), the approach outlined here rewrites (a) to (b) by learning edge directions and weights, then rewrites (b) to (c) via topological sort. Given examples of dependency trees and their corpus-attested surface orders, a neural network can learn to convert previously unseen dependency trees into surface orders by way of a weighted poset.

Implemented as a Graph Neural Network, the machine-learning algorithm treats inputs, targets, and outputs as directed graphs. Further, by representing words with their dependency-based embeddings—that is, embeddings trained on syntactic rather than linear contexts—the model generates a linearized surface order as the final step only, performing all other analysis within a graph framework. In this way surface order is treated as an emergent consequence of topologically sorting an edge-weighted poset, the weights of which represent learned dependency distances.

¹Validity here should be taken neither grammatically nor prescriptively, but rather as a stand-in for attested in a corpus. That is, the model developed herein learns to order the words in a dependency tree based on the structural regularities of a corpus, not intuitive or prescribed grammaticality judgments.

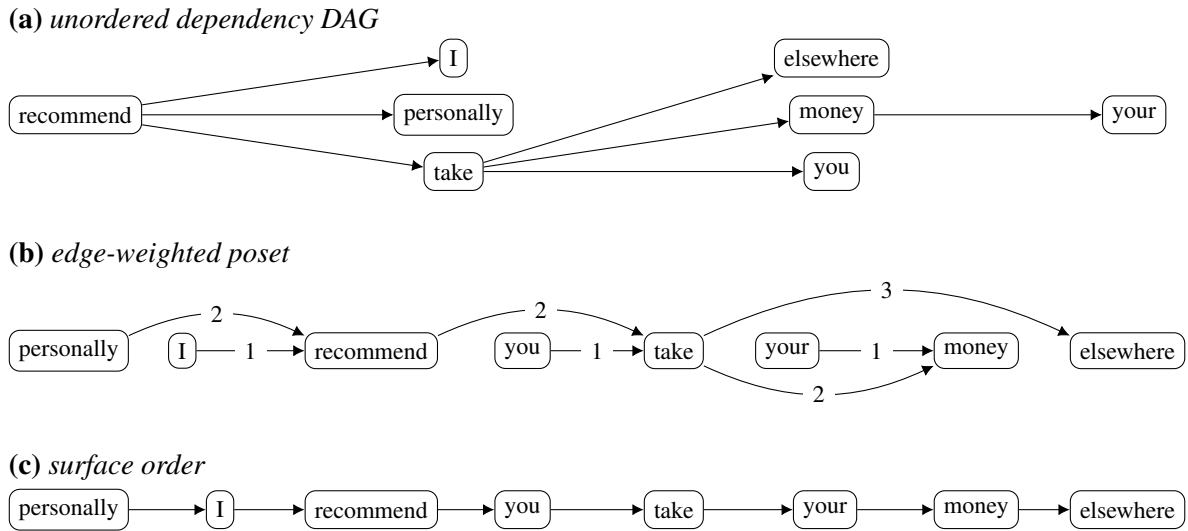


Figure 1: Three graph-theoretic representations of a sentence. (a) A dependency tree as an unordered directed acyclic graph (DAG). (b) A poset in which edge weight indicates dependency distance in the surface order. (c) A surface order generated by a topological sort of the poset in (b).

2 Background literature

2.1 Related linguistic work

Word order is one of the oldest and most prominent areas in the field of linguistics, and as such a wide variety of models have been advanced seeking to describe and understand word-order variation (Song, 2012). It has been approached from generalist perspectives, as in Behaghel’s “what belongs together semantically is also placed close together” (1932, p. 4) or Uniform Information Density (Jaeger and R. Levy, 2006), as well as from specific constituent types, such as the ordering of adpositions and adverbials by manner, place and time (Boisson, 1981); demonstratives, numerals, and descriptive adjectives (Greenberg, 1963; Dryer, 2009); or adjectives by size, shape, and so on (Scott, 2002).

Building on principles such as Head Proximity (Rijkhoff, 1986), Early Immediate Constituents (Hawkins, 1994), Dependency Locality Theory (Gibson, 2000), and Minimize Domains (Hawkins, 2004), a recent approach to word order holds that the dependency distance²—the number of words intervening between a dependent and its head—should be minimized and long-distance dependencies should be avoided (Hudson, 1995; H. Liu et al., 2017). Dependency Distance Minimization (DDM) proposes that a surface order with a smaller cumulative or mean dependency distance is generally favored over alternatives, a tendency that may be universal (Futrell et al., 2015).

However, DDM alone cannot fully explain word order: it does not distinguish between total mirror orders—the dependency distances of *the cat purrs* are presumably the same as *purrs cat the*—or, more plausibly, partial mirror orders such as the swapped adjectives in *big red barn* or *red big barn*. Methods for extending DDM include employing phonemes or syllables as the unit of distance (Ferrer-i-Cancho, 2017), or exploring the relationship between dependents and heads in information-theoretic terms (Dyer, 2018; Hahn et al., 2018). Another avenue is some sort of linear principle that could operate to differentiate mirror surface orders, such as “old concepts come before new ones” (Behaghel, 1932, p. 4), or the possibly contradictory “provide the most important information first” (Gundel, 1988, p. 229).

One way to conceive of surface order is as the result of rewriting a dependency graph by modifying its edge directions to reflect linear order. This process represents an intermediate stage between syntactic structure and surface order in which the linear order of certain word pairs is expressed as a series of precedence relations (Gerdes and Kahane, 2001; Kahane and Lareau, 2016). These precedence relations form a partially ordered set (poset) which can be topologically sorted into a non-unique linearization.

²Dependency distance is also referred to as dependency length in the literature.

2.2 Related NLG work

The field of natural language generation (NLG) seeks to model word order in the service of generating accurate natural language. Contra Harris (1954)³, language is often seen within NLG as a bag of words in which the task of realizing surface order is based on an n -gram language model (Filippova and Strube, 2009). A common implementation follows the bottom-up insights from dependency parsing (Y. Liu et al., 2015), and features such as syntactic category or dependency relations can improve algorithms for linearizing a bag of words (Zhang and Clark, 2015).

The First Multilingual Surface Realisation Shared Task (SR ‘18) brought together nine submissions in a shallow track requiring teams to determine word order and inflections of shuffled and lemmatized Universal Dependencies (UD) data, evaluated by both statistical and human assessment (Mille et al., 2018). For the linearization subtask, of the four submissions with the highest BLEU⁴ scores in at least one of the 10 supported languages: Puzikov and Gurevych (2018) use a bigram language model with binary neural-net classification; Elder and Hokamp (2018) treat the task as a machine-translation problem, using sequence-to-sequence models augmented with synthetic and outside data; Castro Ferreira et al. (2018) sort dependents into preceding and following groups which are then sorted by syntactic category or with a maximum entropy classifier; and King and White (2018) use features such as syntactic category, projectivity, and dependency distance to build a language model to incrementally linearize words.

It has long been noted that a reliance on statistical n -gram metrics like BLEU for measuring generated language is problematic given their inability to generalize seemingly unimportant word order variation or synonymy (Pastra and Saggion, 2003; Turian et al., 2006), as well as their lack of correlation with human assessment (Novikova et al., 2017). BLEU specifically has been criticized given its understudied technological biases, a sufficient reason to avoid using it alone to report scientific evidence (Reiter, 2018, p. 399). Further, while the target or reference of generated language is not necessarily a single sentence—there may be more than one semantically and syntactically valid surface realization of a given set of words, with context determining appropriateness—limited resources often result in a single human-produced reference being used, usually in the guise of an attested sentence in a corpus.

2.3 Projectivity

Projectivity refers to the constraint that a head and its dependents must occur in a contiguous sequence in the surface order (Marcus, 1965). Violations of projectivity—often referred to as discontinuities in the linguistics literature—are instances when a word occurring between a head h and dependent d is not dominated by h in the dependency tree. In the oft-cited non-projective sentence *The hearing is scheduled on the issue today*, both *is* and *scheduled* occur between *hearing* \rightarrow *issue*⁵, but are not dominated by *hearing*. A projective order would be *The hearing on the issue is scheduled today*.

It seems that all natural languages contain some amount of non-projective dependency relations, though calculating exact rates of non-projectivity is difficult given design decisions in the original parsing to create corpora. That is, some annotation schemes presuppose projectivity, and as a result corpora produced following those schemes will not exhibit discontinuities (Ferrer-i-Cancho and Gómez-Rodríguez, 2016). Observed percentages of non-projectivity range from single digits to the mid-teens depending on language, though sources disagree, likely due to differences in corpora, genre, and annotation scheme.

Non-projectivity must be accounted for in any model of word order. Parsers have been developed which allow pseudo-projective (Nivre and Nilsson, 2005), non-projective (Nivre, 2009), and mildly non-projective dependencies (cf. Gómez-Rodríguez, 2016). Similarly, the submissions to SR ‘18 vary with regard to projectivity: of the eight, three explicitly exclude non-projective arcs due to algorithmic design (Basile and Mazzei, 2018; Puzikov and Gurevych, 2018; Sobrevilla Cabezudo and Pardo, 2018), while one follows the tendency toward limited non-projectivity by “encourag[ing] the model to learn that most choices should yield continuous phrases” (King and White, 2018, p. 42).

³“[L]anguage is not merely a bag of words but a tool with particular properties which have been fashioned in the course of use” (p. 156).

⁴BLEU, for bilingual evaluation understudy (Papineni et al., 2002), is “the geometric mean of the n -gram precisions between generated text and reference texts and adds a brevity penalty for shorter sentences” (Mille et al., 2018, p. 4).

⁵Following the UD convention of adpositions depending on nouns, we have *hearing* \rightarrow *issue* and *issue* \rightarrow *on*.

The causal relationship between Dependency Distance Minimization and projectivity is unsettled. Ninio (2017) concludes that “projectivity appears to be not so much a side-effect of DDM as a mathematical requisite for a method to encode a two-dimensional tree in a one-dimensional sentence-string in a way that makes reconstruction possible” (p. 216), appealing to other linguistic structures such as catenae (Osborne et al., 2012) to explain discontinuities. This traditional view—that projectivity exists as a principle independent of DDM—is largely disproven by an analysis which positively correlates dependency distance and the number of crossing dependencies across a variety of multilingual corpora (Ferrer-i-Cancho and Gómez-Rodríguez, 2016). Park and R. Levy (2009) note that an avoidance of long-distance dependencies can result in non-projective surface orders.

2.4 Syntactic word embeddings

The relationship between words has long been thought of distributionally; as Firth (1957) memorably puts it: “you shall know a word by the company it keeps” (p. 11). The company or context of a word is often conceived in terms of the linear neighbors that commonly occur around that word, a context that can be quantified with a dense vector or series of numbers called an embedding. Algorithms have been developed to learn a word’s embedding in a corpus, such as skip-grams (Mikolov et al., 2013). O. Levy and Goldberg (2014) extend the notion of context beyond linear neighbors in their `word2vec-f` to use dependency relations in learning syntactic embeddings: a word’s context is based on the heads and dependents it takes in a corpus.

The number of dimensions necessary for a given task is an understudied problem. It is widely accepted that larger dimensions are better, up to a point of diminishing returns; for example, O. Levy and Goldberg (2014) use 300 in their evaluation, mentioning that 600 produces similar results. However, Spirling and Rodriguez (2019) note that very large dimensions relative to corpus size result in greater instability of embeddings, where instability refers to the rate at which the cosine-similar nearest neighbors differ between models (Wendlandt et al., 2018). Patel and Bhattacharyya (2017) explore the lower bound of embedding dimensions, below which performance suffers, providing a rather complicated method for calculating the minimum based on the maximum clique of a cosine-similarity matrix of word co-occurrence. An industry rule-of-thumb⁶ is to use the fourth root of vocabulary size.

2.5 Graph neural networks

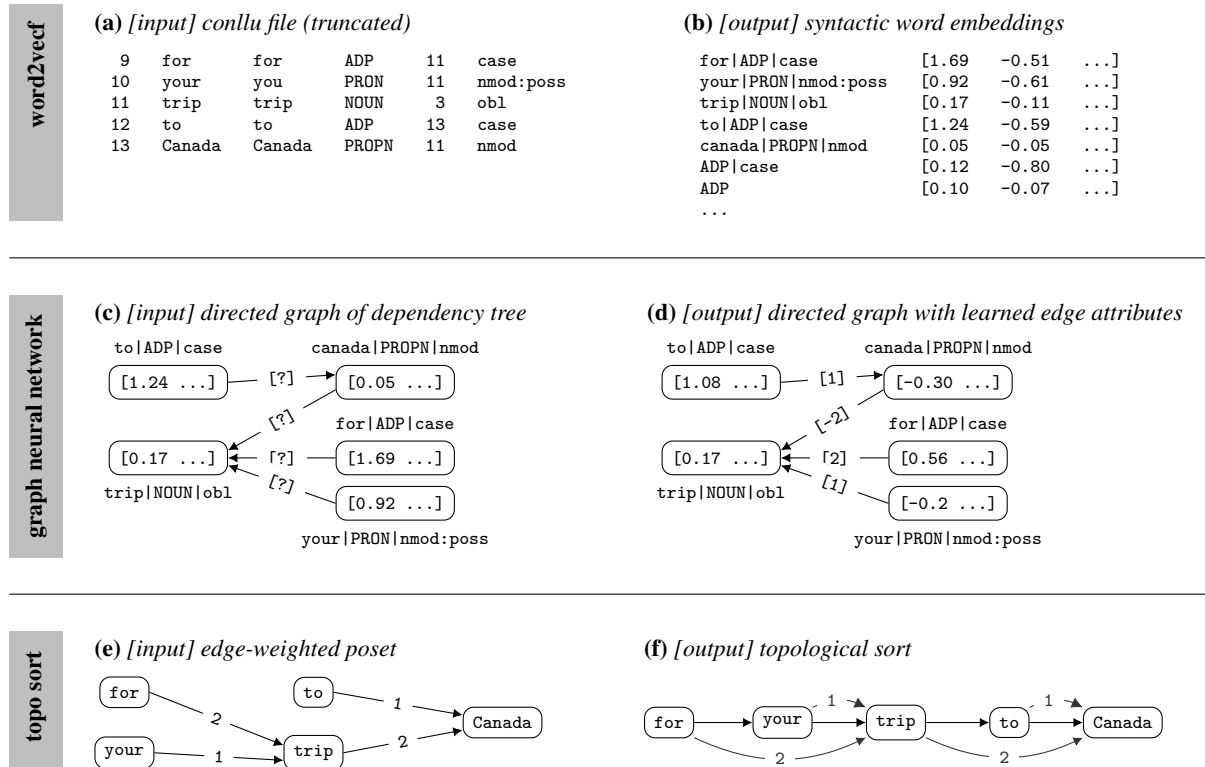
While machine-learning algorithms, deep or otherwise, have traditionally operated on data represented in Euclidean space—for example, image data can be represented as a regular grid of pixel values—graph neural networks (GNN) allow the complexity of graph structures to be analyzed (Wu et al., 2019). The Graph Nets (GN) framework relies on a graph-to-graph model called a GN block “which takes a graph as input, performs computations over the structure, and returns a graph as output” (Battaglia et al., 2018, p. 11). In this framework, a graph is composed of nodes and their attributes, edges and their attributes, and a set of global attributes. Input and target graphs may contain different node and edge configurations; only the attributes for nodes and the attributes for edges must be of a consistent form. It is these sets of attributes which form the learned parameters of the neural network.

GN blocks also support message-passing neural networks (MPNN) (Gilmer et al., 2017), a method by which a graph’s node and edge attributes undergo spatial-based graph convolutions and pooling (Wu et al., 2019, p. 8). In this manner a graph’s connected nodes influence each other’s node and edge attributes, passing information along directed edges.

3 Methodology

The approach described in the current study rests on the notion that adding dependency distances as positive or negative edge weights to a dependency tree allows the DAG to be rewritten as a poset whose topological sort reflects a surface order. Edge weights are therefore the number of words intervening between a dependent and its head, where negative weights indicate a dependent that precedes its head and positive a dependent following its head. Learning these edge weights is the core goal of the model.

⁶<https://developers.googleblog.com/2017/11/introducing-tensorflow-feature-columns.html>



3.2 Graph neural network implementation

The machine-learning algorithm is implemented using Graph Nets and Sonnet, two DeepMind libraries⁹ for building graph neural networks using Google’s Tensorflow (Abadi et al., 2015). The network’s layers contain 18 neurons each and follow an ‘encode-process-decode’ model common to many Graph Nets implementations. Because learned edge weights in the GNN can be positive or negative, loss is calculated as the absolute difference between target and output. An Adam optimizer with a learning rate of 1^{-3} is used, there are 6 message-passing steps, and the network is run through 10,000 iterations.

The input is a series of networkx¹⁰ directed graphs, one for each sentence in the training and testing sets. In order to effectively utilize message passing, edges are constructed as *dependent* \rightarrow *head*, opposite the usual syntactic dependency-parsing edge direction. Each node has an attribute which is the vector produced by `word2vecf`’s syntactic embedding. In the GNN, edge weights are used to track dependency distance, both negative and positive. A negative edge weight indicates that a head precedes its dependent, and a positive weight that a dependent precedes its head. Target edge weights are calculated as the difference between the dependent and head location in the original surface order, normalized to $[-1,1]$ by dividing each distance by the maximum dependency distance of a given sentence.

For example, Figure 2 (c) and (d) show the input and output for the phrase *for your trip to Canada*. The input to the GNN is the dependency tree, where each node’s attribute is the word’s syntactic embedding. The output is the same dependency tree with learned edge attributes reflecting the distance between dependent and head.

3.3 Weighted topological sort

Performing a topological sort of an edge-weighted poset such that connected nodes are placed in ascending order of edge weight is conceptually quite simple, but implementation is more complicated than it may appear. A straightforward approach of simply merging nodes with the smallest weights before those with larger weights does not properly order the nodes, since the weight of arcs crossing the merged nodes are not necessarily updated to reflect the merge. Instead, as outlined in Algorithm 1, each edge (u, v) from the poset can be added to a new directed graph *order* such that the edge’s weight is maintained, even though u and v may not be adjacent in *order*.

When inserting edge (u, v) with weight w_{uv} into *order*, if u is already in *order*, then traverse the successor nodes of u until the total distance from u —a value maintained by w_{sum} —exceeds w_{uv} . At that point, insert v and update the weights of v ’s neighbors. This process is shown in lines 5-16. Similarly, as shown in lines 17-28, if v is already in *order*, traverse the predecessor nodes of v until w_{sum} exceeds w_{uv} , insert u , and update u ’s neighbors’ weights. Finally, if neither u nor v are in *order*, add edge (u, v) with weight w_{uv} to *order*, as shown in line 30. When all edges from *poset* have been added to *order*, the topological sort of *order* is returned as the surface realization. Each edge in *poset* must be added to *order*, and in the worst-case scenario the weight of each existing edge in *order* must be examined. Therefore Algorithm 1 runs in $O(n \log n)$ time, where n is the number of edges in *poset*.

3.4 Baseline (AVG)

Rather than generating syntactic word embeddings and running the GNN, a naive approach to determining dependency distances is to average the distance between any two words in the training set for use on the testing set. Similar to the set of word embeddings (§3.1), in order to generalize to unseen words in the test set, average distances are created for each pair dependent pair of word|POS|relation, POS|relation, and POS. For example, if the |DET|det has an average dependency distance of 1.2 from horse|NOUN|nsubj, and brown|ADJ|amod has an average of 0.9 from horse|NOUN|subj, then using those two average distances as weights in a poset would result in a surface order of *the brown horse*. If red|ADJ|amod was unseen during training, then the average of all instances of ADJ|amod dependent on horse|NOUN|subj would be used—if that average distance were 1.3, then this naive approach would return *red the horse*.

⁹<https://github.com/deepmind/>

¹⁰<https://networkx.github.io/>

Algorithm 1: Given an edge-weighted *poset*, construct a total order such that nodes with smallest weights are adjacent.

```

1:  function WEIGHTED_TOPO_SORT(poset)
2:    order  $\leftarrow \emptyset$                                  $\triangleright$  empty directed graph to hold totally ordered set
3:    for  $(u, v, w_{uv}) \in \textit{poset}$  do
4:       $w_{sum} \leftarrow 0$                                  $\triangleright$  a sum of traversed weights
5:      if  $u \in \textit{order}$  then
6:        while  $w_{uv} > w_{sum}$  do                         $\triangleright$  traverse successors of  $u$ 
7:           $s \leftarrow \textit{order}.u.\textit{successor}$ 
8:           $w_{us} \leftarrow \textit{order}[u][s].\textit{weight}$ 
9:           $w_{sum} \leftarrow w_{sum} + w_{us}$ 
10:         if  $w_{uv} < w_{sum}$  then
11:            $u \leftarrow s$                                  $\triangleright u$  becomes its successor  $s$ 
12:         end if
13:       done
14:        $w_{vs} \leftarrow w_{sum} - w_{uv}$                      $\triangleright w_{vs}$  is how much  $w_{sum}$  overshot  $w_{uv}$ 
15:       order.UPDATE_EDGE( $u, s, \_$ )  $\leftarrow$                  $\triangleright$  change existing  $(u, s)$ ...
16:          $[(u, v, w_{us} - w_{vs}), (v, s, w_{vs})]$            $\triangleright$  ... to  $(u, v)$  and  $(v, s)$  and update weights
17:     else if  $v \in \textit{order}$  then
18:       while  $w_{uv} > w_{sum}$  do                         $\triangleright$  traverse predecessors of  $v$ 
19:          $p \leftarrow \textit{order}.v.\textit{predecessor}$ 
20:          $w_{pv} \leftarrow \textit{order}[p][v].\textit{weight}$ 
21:          $w_{sum} \leftarrow w_{sum} + w_{pv}$ 
22:         if  $w_{uv} < w_{sum}$  then
23:            $v \leftarrow p$                                  $\triangleright v$  becomes its predecessor  $p$ 
24:         end if
25:       done
26:        $w_{pu} \leftarrow w_{sum} - w_{uv}$                      $\triangleright w_{pu}$  is how much  $w_{sum}$  overshot  $w_{uv}$ 
27:       order.UPDATE_EDGE( $p, v, \_$ )  $\leftarrow$                  $\triangleright$  change existing  $(p, v)$ ...
28:          $[(p, u, w_{pu}), (u, v, w_{pv} - w_{pu})]$            $\triangleright$  ... to  $(p, u)$  and  $(u, v)$  and update weights
29:     else
30:       order.ADD_EDGE( $u, v, w_{uv}$ )
31:     end if
32:   done
33:   return TOPO_SORT(order)                                $\triangleright$  return topological sort of order graph
34: end function

```

3.5 Evaluation

To evaluate the performance of the GNN algorithm compared to the AVG baseline in an automated way across various languages, we must unfortunately use a single target reference to compare the generated sentences. Thus the reference for each sentence is the attested version in the source UD corpus; the generated sentences from both AVG and GNN will be measured for similarity to the attested version.

The algorithm is attempting to order a set of words as closely as possible to their original surface realization in the corpus. Because words may repeat in the sentence, each order is instead represented by a list of integers, and it is these lists of integers which are compared. For example, assuming a target reference order of $[1, 2, 3]$ for *the red horse*, the generated order of *red the horse* would be $[2, 1, 3]$. An obvious way to quantify how similar these integer lists are is with the widely used Spearman’s rank correlation coefficient (Spearman, 1904), also known as Spearman’s ρ (rho), which non-parametrically measures the similarity of two rankings. It ranges from -1, indicating that one order is the reverse of the other, to 1, for perfect correlation. The example of $[1, 2, 3]$ $[2, 1, 3]$ returns a ρ of 0.5, since in the second order 1 and 2 both precede 3, but 1 does not precede 2. This measure tells us which approach, AVG or GNN, generates orders closest to the attested UD order, as well as a loose gauge of overall effectiveness for both the general approach as well as each algorithm.

Further, to address the question of projectivity, the percentage of projective dependency arcs generated by the AVG baseline, the GNN algorithm, and the attested sentences is evaluated. In each case, projectivity is calculated as the number of instances in which a word appearing between a head h and dependent d is not dominated by h . This measure allows us to explore how dependency distance might result in known rates of projectivity in natural language.









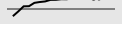

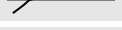

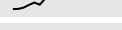





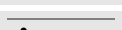







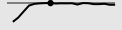


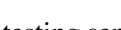
					SPEARMAN'S ρ [-1,1]			PROJECTIVITY [0,1]			
		N _{TR}	N _{TE}	D _V	AVG		GNN	AVG		GNN	UD
Afrikaans	AfriBooms	1315	425	18	0.707		0.773	0.530		0.650	0.939
Armenian	ArmTDP	560	470	18	0.628		0.672	0.413		0.585	0.987
Czech	CLTT	755	121	18	0.665		0.659	0.359		0.469	0.982
English	ParTUT	1781	153	20	0.634		0.775	0.496		0.680	0.995
French	ParTUT	803	110	18	0.677		0.729	0.531		0.669	0.998
Greek	GDT	1632	450	22	0.731		0.754	0.503		0.651	0.996
Hungarian	Szeged	910	449	20	0.635		0.609	0.440		0.598	0.969
Irish	IDT	566	452	18	0.674		0.753	0.461		0.603	0.978
Italian	ParTUT	1781	153	22	0.657		0.796	0.482		0.651	0.996
Latin	Perseus	1334	939	20	0.614		0.582	0.613		0.729	0.855
Maltese	MUDT	1119	516	18	0.729		0.750	0.498		0.682	0.995
Slovenian	SST	1669	890	18	0.549		0.567	0.663		0.798	0.967
Telugu	MTG	1051	146	14	0.916		0.931	0.925		0.971	0.997
Uyghur	UDT	1656	900	20	0.728		0.727	0.629		0.762	0.976

Table 1: Results. Each language is listed by its corpus; number of training and testing sentences; embedding dimension; Spearman’s ρ rank correlation coefficient for AVG and GNN; and rate of projectivity for AVG, GNN, and as attested in the UD corpus. Boldfaced numbers indicate cases in which GNN performed better than AVG. Sparklines show trends over 10K iterations with horizontal gray lines indicating AVG performance and black dots showing peak performance of GNN.

4 Results & Discussion

Table 1 shows the results of running both the AVG baseline and GNN algorithm on 14 v2.4 UD corpora representing a range of language families. These are relatively small corpora—between 500 and 2000 training sentences—and as a consequence their small vocabularies result in embedding vector dimensions between 14 and 22 due to the use of twice the fourth root of vocabulary size (§2.4, §3.1). While smaller than the more usual 50- or 300-element vectors, tying dimensionality to corpus vocabulary size seemed to avoid instability in the embedding space, though perhaps not in every case. Further, experiments with larger dimensions resulted in poor generalization to the testing set, possibly due to a lack of correlation between embeddings seen and unseen during training.

Results from Spearman’s ρ rank correlation show that both AVG and GNN were able to positively correlate surface order with the source UD corpora. Because Spearman’s ρ ranges from -1 to 1, positive values are better than chance; values above 0.5 seem rather promising. A large part of surface order can apparently be predicted based on dependency distance, averaged or learned. In all cases the GNN was able to approach AVG, exceeding it 10 out of 14 times. For many languages, the GNN achieved its peak value before training was complete, probably indicating overfitting. In the cases in which the GNN did not best AVG, the sparkline trends for Czech, Hungarian, and Latin suggest problems during training, perhaps due to overzealous learning rates or unstable embeddings, while Uyghur came very close.

In terms of projectivity, the GNN outperformed AVG in all cases, even when it did not best AVG in terms of Spearman’s ρ . While it is of course true that were the AVG or GNN method able to perfectly capture the word order of the UD corpora, the rate of projectivity and Spearman’s ρ would match exactly, but it is intriguing that short of perfection, Spearman’s ρ and projectivity are not necessarily correlated. Nor do many of the intralanguage trends match between the two measures—the highest GNN projectivity was generally achieved late in the training process, and the two sparklines of, for example, Armenian, are not very similar. While the GNN outperformed AVG in generating surface orders with higher rates of projectivity, even those rates lagged quite a bit behind the actual rates for almost all languages. This is likely due to even seemingly minor word transpositions leading to non-projective arcs (§4.1).

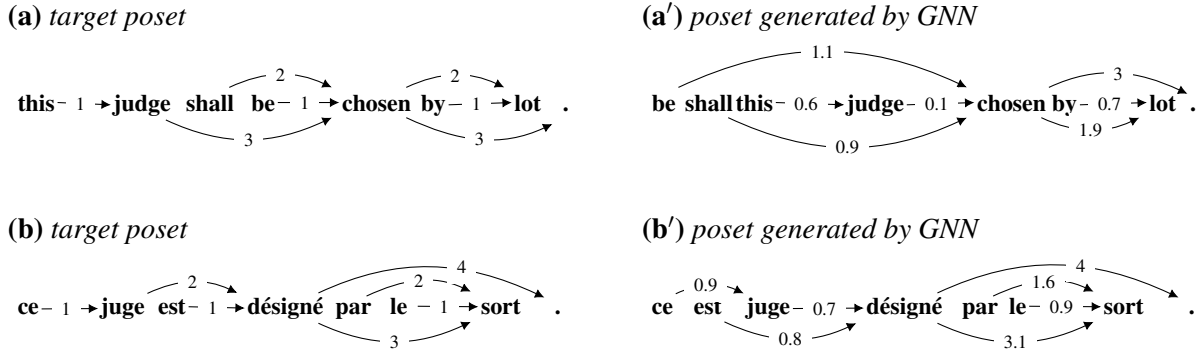


Figure 3: Target and generated posets from English- and French-ParTUT corpora.

Importantly, AVG is a naive approach, not a learning algorithm. As such there is very little room for improvement by adjusting how the averaged dependency distances are determined—employing morphological data or using lemmata instead of wordforms, for example. Conversely, changes to number of iterations, architecture, or hyperparameters of the GNN, especially tailored to each corpus, would almost certainly yield even better results, with a hypothetical upper bound limited only by the irreducible error present in a language’s word-order variation.

The results confirm that dependency distances can be learned from dependency trees by the GNN algorithm, usually better than a naive approach. Those distances can be used to generate surface realizations with word orders that positively correlate with attested UD sentences. Because these promising results can be generated from an essentially off-the-shelf GNN with relatively standardized parameters across a wide variety of languages, future endeavors improving the GNN architecture is certainly warranted.

4.1 Error analysis

Delving a bit into the sorts of errors in the surface orders generated by GNN, Figure 3 shows four versions of the same sentence: (a) the poset for a sentence from the UD English-ParTUT corpus; (a’) the poset generated by GNN with a Spearman’s ρ coefficient of 0.786—only slightly higher than the average ρ for that corpus, and therefore a typical generation; (b) the poset for the same sentence from French-ParTUT; and (b’) the poset generated by GNN with a non-projective¹¹ arc.

Figure 3 (a’) deviates from (a) in that the weight of $be \xrightarrow{1.1} chosen$ is larger than $shall \xrightarrow{0.9} chosen$, and both those edges have weights larger than the combination of $this \xrightarrow{0.6} judge$ and $judge \xrightarrow{0.1} chosen$. The result is a sentence in which the auxiliaries *be* and *shall* are transposed, and both appear in front of *this judge*. Similarly, (b’) deviates from (b) in that the weight of $est \xrightarrow{0.8} désigné$ is larger than the weight of $juge \xrightarrow{0.7} désigné$ —though unlike the English not larger than the combined weights of $ce \xrightarrow{0.9} juge$ and $juge \xrightarrow{0.7} désigné$. The result is a transposition of *juge* and *est*, causing a non-projective arc as *est* appears between *ce* and *juge* but is not dominated by either.

Aside from the transposition of the auxiliaries in (a’), both generated surface orders suffer from the weight of $judge/juge \rightarrow chosen/désigné$ being too small. While the offending edge in (a’) is quite small at 0.02, requiring an addition of over 2 to overcome the combined weights of the auxiliaries *be* and *shall*, an addition of just 0.11 to the weight of the edge would resolve (b’). In other words, if the weight of $est \xrightarrow{0.8} désigné$ were increased to 0.81, it would be larger than $est \xrightarrow{0.8} désigné$ and therefore *juge* would precede *est*, resolving (b’) to (b).

Neither training set for these corpora contains the word *judge/juge*, so the word’s embedding collapses to an average of all nouns acting as passive subjects, NOUN|nsubj:pass. This suggests that insufficient training size, lack of proper generalization from the available training data, and/or problematic embedding creation for unseen words is at fault here. These can all be addressed in future research.

¹¹The graphs in Figure 3 are posets, not dependency trees, and therefore the dependency concept of projectivity is not readily apparent. A poset analogue is planarity in the half-plane, or 1-planarity (cf. Pitler et al., 2013, p. 19). If all arcs in Figure 3 (b’) were drawn above the words, we would see that the $ce \rightarrow juge$ and $est \rightarrow désigné$ arcs would cross.

4.2 Dependency distance tolerance & projectivity

What is being learned by the GNN? That is, what do the edge weights, used to create a poset, actually represent? The question is perhaps conceptually a bit easier with AVG: the weights are the average distances between dependents and their heads in a corpus. AVG calculates how far a dependent tends to be from its head, or put another way, how many intervening words tend to be allowed between dependent and head in a collection of surface orders. It is a dependent word’s tolerance for how far it can be placed in front or behind its head in a surface realization. It seems that the GNN is learning this same information about dependency distance tolerance, but in a more subtle and context-sensitive way. Rather than simply an average distance, the GNN is learning how far a dependent can be placed from its head in concert with its syntactically related words¹² in a given dependency tree.

Dependency distance tolerance is effectively a maximum for how far apart a dependent and head can be in the surface realization of a given dependency tree. What factors determine this tolerance and how it might be encoded in a linguistic system is left for other research. However, dependency distance tolerance is a useful concept for exploring how projectivity might come about.

It was suggested in §2.3 that observed rates of projectivity might emerge from Dependency Distance Minimization (DDM). That is, the desire to minimize cumulative or mean dependency distances results in the high rates of projectivity seen across languages. A further goal within DDM is to avoid long-distance dependencies, though this avoidance may result in non-projective surface orders. The concept of dependency distance tolerance provides a more nuanced view of this second DDM motivation.

The topological sort of a poset whose edge weights correspond to contextual dependency tolerances, at least as implemented here, may place dependents closer to their heads than their tolerance, but not farther. As such, it defines an upper bound for each edge weight in a poset. A surface order can be seen as the result of assembling words such that dependents are placed no farther from their heads than their tolerance. In this way dependency distances in the surface order are not only minimized, but minimized in such a way that each word’s contextual dependency tolerance is taken into account.

Thus the topological sort of a weighted poset implements DDM’s goal of minimizing dependency distances generally, while the learned dependency tolerances provide a contextually sensitive definition of what ‘long distance’ means for each dependent pair in order to avoid generating surface orders with long-distance dependencies. Through this lens both the strong tendency towards projectivity across languages, as well as the occasional instances of non-projectivity, can be seen as an effect of avoiding dependency distances which exceed their contextual tolerances.

5 Summary

This paper describes a novel method for converting dependency trees to surface orders via syntactic word embeddings and edge-weighted posets. The embeddings are learned via `word2vecf`, and poset edge directions and weights are learned by a graph neural network (GNN), all trained on Universal Dependencies (UD) corpora. An algorithm is provided for topologically sorting a weighted poset. The output of the GNN is compared to a naive baseline in which average dependency distances are used as poset edge weight, both evaluated against attested word orders in UD corpora representing a variety of language families. The GNN outperforms the baseline on 10 of 14 corpora in terms of rank correlation and in all cases in terms of rate of projectivity.

The main contribution of the paper is the insight that a surface order can be represented by an edge-weighted poset, the weights of which can be learned by a graph neural network. Representing surface order as the result of topologically sorting this poset contributes to our understanding of how a tendency towards projectivity across natural languages might be explained.

Future research directions include improvement of the GNN architecture and hyperparameters; exploration of the interaction between word embedding dimension, performance, and generalizability; and the analysis of larger corpora.

¹²Due to the graph nature of the GNN, message passing, and the use of syntactic embeddings, a word’s context for determining dependency distance in this study is entirely dependency based, never linear.

References

- Martin Abadi et al. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. URL: <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45166.pdf>.
- Valerio Basile and Alessandro Mazzei (2018). The DipInfo-UniTo system for SRST 2018. *Proceedings of the First Workshop on Multilingual Surface Realisation*. Melbourne, Australia: Association for Computational Linguistics, pp. 65–71.
- Peter W. Battaglia et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261 [cs, stat]*.
- Otto Behaghel (1932). *Deutsche Syntax eine geschichtliche Darstellung*. Heidelberg: Carl Winters Universitätsbuchhandlung.
- Claude Boisson (1981). Hiérarchie universelle des spécifications de temps, de lieu, et de manière. *Confluents* 7, pp. 69–124.
- Thiago Castro Ferreira, Sander Wubben, and Emiel Krahmer (2018). Surface Realization Shared Task 2018 (SR18): The Tilburg University Approach. *Proceedings of the First Workshop on Multilingual Surface Realisation*. Melbourne, Australia: Association for Computational Linguistics, pp. 35–8.
- Matthew S. Dryer (2009). On the order of demonstrative, numeral, adjective, and noun: an alternative to Cinque. *Conference on theoretical approaches to disharmonic word orders*.
- William Dyer (2018). Integration complexity and the order of cosisters. *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*. Brussels, Belgium: Association for Computational Linguistics, pp. 55–65.
- Henry Elder and Chris Hokamp (2018). Generating High-Quality Surface Realizations Using Data Augmentation and Factored Sequence Models. *Proceedings of the First Workshop on Multilingual Surface Realisation*. Melbourne, Australia: Association for Computational Linguistics, pp. 49–53.
- Ramon Ferrer-i-Cancho (2017). Towards a theory of word order. Comment on "Dependency distance: a new perspective on syntactic patterns in natural language" by Haitao Liu et al. *Physics of Life Reviews*.
- Ramon Ferrer-i-Cancho and Carlos Gómez-Rodríguez (2016). Crossings as a side effect of dependency lengths. *Complexity* 21 (S2), pp. 320–328.
- Katja Filippova and Michael Strube (2009). Tree Linearization in English: Improving Language Model Based Approaches. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Boulder, Colorado: Association for Computational Linguistics, pp. 225–8.
- John Rupert Firth (1957). A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis*. Oxford: Philological Society, pp. 1–32.
- Richard Futrell, Kyle Mahowald, and Edward Gibson (2015). Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences* 112.33, pp. 10336–41.
- Kim Gerdes and Sylvain Kahane (2001). Word Order in German: A Formal Dependency Grammar Using a Topological Hierarchy. *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*. Toulouse, France: Association for Computational Linguistics, pp. 220–7.
- Edward Gibson (2000). The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, pp. 95–126.
- Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl (2017). Neural Message Passing for Quantum Chemistry. *arXiv:1704.01212 [cs]*.
- Carlos Gómez-Rodríguez (2016). Restricted Non-Projectivity: Coverage vs. Efficiency. *Computational Linguistics* 42.4, pp. 809–17.

- Joseph Greenberg (1963). Some universals of grammar with particular reference to the order of meaningful elements. *Universals of Grammar*. Ed. by Joseph Greenberg. Cambridge, Massachusetts: MIT Press, pp. 73–113.
- Jeanette Gundel (1988). Universals of topic-comment structure. *Studies in Syntactic Typology*. Ed. by Michael Hammond, Edith Moravcsik, and Jessica Wirth. Philadelphia: John Benjamins Publishing, pp. 209–39.
- Michael Hahn, Judith Degen, Noah Goodman, Dan Jurafsky, and Richard Futrell (2018). An Information-Theoretic Explanation of Adjective Ordering Preferences. *Proceedings of the 40th annual conference of the Cognitive Science Society*. London: Cognitive Science Society.
- Zellig S. Harris (1954). Distributional Structure. *WORD* 10.2, pp. 146–62.
- John A. Hawkins (1994). *A Performance Theory of Order and Constituency*. Cambridge: Cambridge University Press.
- John A. Hawkins (2004). *Efficiency and Complexity in Grammars*. Oxford: Oxford University Press.
- Richard Hudson (1995). Measuring syntactic difficulty. URL: <http://dickhudson.com/wp-content/uploads/2013/07/Difficulty.pdf>.
- T. Florian Jaeger and Roger Levy (2006). Speakers optimize information density through syntactic reduction. *Advances in neural information processing systems*, pp. 849–56.
- Sylvain Kahane and François Lareau (2016). Word Ordering as a Graph Rewriting Process. *Formal Grammar*. Ed. by Annie Foret, Glyn Morrill, Reinhard Muskens, Rainer Osswald, and Sylvain Pogodalla. Springer Berlin Heidelberg, pp. 216–39.
- David King and Michael White (2018). The OSU Realizer for SRST ‘18: Neural Sequence-to-Sequence Inflection and Incremental Locality-Based Linearization. *Proceedings of the First Workshop on Multilingual Surface Realisation*. Melbourne, Australia: Association for Computational Linguistics, pp. 39–48.
- Omer Levy and Yoav Goldberg (2014). Dependency-Based Word Embeddings. *ACL (2)*. Citeseer, pp. 302–8.
- Haitao Liu, Chunshan Xu, and Junying Liang (2017). Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of Life Reviews* 21, pp. 171–93.
- Yijia Liu, Yue Zhang, Wanxiang Che, and Bing Qin (2015). Transition-Based Syntactic Linearization. *HLT-NAACL*.
- Solomon Marcus (1965). Sur la notion de projectivité. *Mathematical Logic Quarterly* 11.2, pp. 181–92.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). Efficient estimation of word representations in vector space. *arXiv:1301.3781 [cs]*.
- Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner (2018). The First Multilingual Surface Realisation Shared Task (SR’18): Overview and Evaluation Results. *Multilingual Surface Realisation: Shared Task and Beyond: Proceedings of the Workshop*. Multilingual Surface Realisation: Shared Task and Beyond. Melbourne, Australia: Association for Computational Linguistics, pp. 1–12.
- Anat Ninio (2017). Projectivity is the mathematical code of syntax. *Physics of Life Reviews* 21, pp. 215–7.
- Joakim Nivre (2009). Non-projective dependency parsing in expected linear time. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, pp. 351–9.
- Joakim Nivre and Jens Nilsson (2005). Pseudo-projective dependency parsing. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 99–106.
- Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser (2017). Why We Need New Evaluation Metrics for NLG. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2231–40.

- Timothy Osborne, Michael Putnam, and Thomas Groß (2012). Catenae: Introducing a Novel Unit of Syntactic Analysis. *Syntax* 15.4, pp. 354–96.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). Bleu: a Method for Automatic Evaluation of Machine Translation. *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 311–18.
- Y. Albert Park and Roger Levy (2009). Minimal-length linearizations for mildly context-sensitive dependency trees. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 335–43.
- Katerina Pastra and Horacio Saggion (2003). Colouring Summaries BLEU. *Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing: Are Evaluation Methods, Metrics and Resources Reusable? Evalinitatives '03*. event-place: Budapest, Hungary. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 35–42.
- Kevin Patel and Pushpak Bhattacharyya (2017). Towards Lower Bounds on Number of Dimensions for Word Embeddings. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, pp. 31–6.
- Emily Pitler, Sampath Kannan, and Mitchell Marcus (2013). Finding Optimal 1-Endpoint-Crossing Trees. *Transactions of the Association for Computational Linguistics* 1, pp. 13–24.
- Alain Polguère and Igor Mel'čuk (2009). *Dependency in Linguistic Description*. John Benjamins Publishing.
- Yevgeniy Puzikov and Iryna Gurevych (2018). BinLin: A Simple Method of Dependency Tree Linearization. *Proceedings of the First Workshop on Multilingual Surface Realisation*. Melbourne, Australia: Association for Computational Linguistics, pp. 13–28.
- Ehud Reiter (2018). A Structured Review of the Validity of BLEU. *Computational Linguistics* 44.3, pp. 393–401.
- Jan Rijkhoff (1986). Word Order Universals Revisited: The Principle of Head Proximity. *Belgian Journal of Linguistics* 1, pp. 95–125.
- Gary-John Scott (2002). Stacked adjectival modification and the structure of nominal phrases. *Functional Structure in DP and IP: The Cartography of Syntactic Structures*. Vol. 1. New York: Oxford University Press, pp. 91–120.
- Marco Antonio Sobrevilla Cabezudo and Thiago Pardo (2018). NILC-SWORNEMO at the Surface Realization Shared Task: Exploring Syntax-Based Word Ordering using Neural Models. *Proceedings of the First Workshop on Multilingual Surface Realisation*. Melbourne, Australia: Association for Computational Linguistics, pp. 58–64.
- Jae Jung Song (2012). *Word Order*. New York: Cambridge University Press.
- Charles Spearman (1904). The Proof and Measurement of Association between Two Things. *The American Journal of Psychology* 15.1, pp. 72–101.
- Arthur Spirling and Pedro L Rodriguez (2019). What works, what doesn't, and how to tell the difference for applied research. URL: <https://www.nyu.edu/projects/spirling/documents/embed.pdf>.
- Lucien Tesnière (1959). *Éléments de syntaxe structural*. Paris: Klincksieck.
- Joseph P. Turian, Luke Shea, and I. D. Melamed (2006). *Evaluation of Machine Translation and its Evaluation*: Fort Belvoir, VA: Defense Technical Information Center.
- Laura Wendlandt, Jonathan K. Kummerfeld, and Rada Mihalcea (2018). Factors Influencing the Surprising Instability of Word Embeddings. *arXiv:1804.09692 [cs]*.
- Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu (2019). A Comprehensive Survey on Graph Neural Networks. *arXiv:1901.00596 [cs, stat]*.
- Yue Zhang and Stephen Clark (2015). Discriminative Syntax-Based Word Ordering for Text Generation. *Computational Linguistics* 41.3, pp. 503–38.

Linked Open Treebanks. Interlinking Syntactically Annotated Corpora in the LiLa Knowledge Base of Linguistic Resources for Latin

Francesco Mambrini, Marco Passarotti

CIRCSE Research Centre

Università Cattolica del Sacro Cuore

Largo Gemelli, 1 - 20123 Milan, Italy

{francesco.mambrini}{marco.passarotti}@unicatt.it

Abstract

In spite of the current availability of large collections of treebanks that can be used and queried from one common place on the web, we are still far from achieving a real interconnection, both between treebanks themselves and with other (kinds of) linguistic resources. However, making resources interoperable is a crucial requirement to maximize the contribution of each single resource, as well as to account for the linguistic complexity of the texts provided by (annotated) corpora and particularly by treebanks. This paper describes how dependency treebanks are interlinked in a Knowledge Base of linguistic resources for Latin based on Linked Open Data practices and standards. The Knowledge base is built to make linguistic resources interact by integrating all types of annotation applied to a particular word/text into a common representation.

1 Introduction and Motivation

Dependency treebanks for Latin have a history that goes back to 2006. For it was in that year that the first two projects kicked off: the Latin Dependency Treebank (LDT) (Bamman and Crane, 2006), featuring a small selection of texts by Classical authors (currently around 50k nodes), and the *Index Thomisticus* Treebank (IT-TB) (Passarotti, 2011), based on works written in the XIIIth century by Thomas Aquinas (approximately 400k nodes). Later on, a third Latin treebank was created in the context of the PROIEL project (Haug and Jøhndal, 2008), which includes the entire New Testament in Latin (the so called *Vulgata* by Jerome) and texts from the Classical era (for a total of around 250k nodes). Most recently, a syntactically annotated corpus of original VIIIth-IXth century charters from Central Italy, called Late Latin Charter Treebank (LLCT; around 250k nodes), was made available (Korkiakangas and Passarotti, 2011). While the LDT, the IT-TB and the LLCT have shared the same manual for syntactic annotation since the beginning of their respective projects (Bamman et al., 2007), the PROIEL treebank follows a slightly different style (Haug, 2010). Currently, all the Latin treebanks except the LLCT are available also in the Universal Dependencies collection (UD) (Nivre et al., 2016).¹

The existence of four treebanks for an ancient language like Latin is not surprising, reflecting the large diachronic (as well as diatopic) span of Latin texts, which are spread across a time frame of more than two millennia and in most areas of the Mediterranean and of what is called Europe today. Since Latin has represented for a long time a kind of *lingua franca*, the variety of its textual typologies is wide, including scientific treaties, literary works, philosophical texts and official documents. This aspect makes it impossible to build one textual corpus that alone can be sufficiently representative of “Latin”, just because there are too many varieties of Latin, which can be even very different from each other.²

In order to cope with such a large variety, several collections of Latin texts are today available in digital format, like for instance the *Perseus Digital Library*³ and the collection of Medieval Italian Latinity *ALIM*.⁴

Besides textual resources, the centuries-old tradition of Latin lexicography resulted in the current availability of several digitized dictionaries, like for instance the Lewis-Short dictionary available at Perseus

¹<http://universaldependencies.org/>

²For instance, Ponti and Passarotti (2016) show the dramatic decrease of the accuracy rates of a dependency parsing pipeline trained on the IT-TB when applied on texts of the Classical era taken from the LDT.

³<http://www.perseus.tufts.edu/hopper/>

⁴<http://www.alim.dfll.univr.it/>

and the *Thesaurus Linguae Latinae* by the Bayerische Akademie der Wissenschaften in Munich.⁵ A small *Latin WordNet* including around 9,000 lemmas is also available (Minozzi, 2010), as well as a derivational morphology lexicon called *Word Formation Latin* (wfl) (Litta et al., 2016).

Just like for most other (both modern and ancient) languages, the interoperability issues imposed by the different formats, tag sets and annotation criteria of the linguistic resources for Latin severely limit their potential for exploitation and use. Indeed, linking linguistic resources to one another would maximize their contribution to linguistic analysis at multiple levels, be those lexical, morphological, syntactic, semantic or pragmatic. Thus, presently there is a growing interest in the interoperability of (annotated) corpora, lexical resources and Natural Language Processing (NLP) tools (Ide and Pustejovsky, 2010). So far, this was partially approached by building large infrastructures and databases of linguistic resources, like CLARIN,⁶ DARIAH,⁷ META-SHARE,⁸ and EAGLE.⁹ In the treebank area, the UD collection includes more than 100 treebanks sharing the same annotation guidelines and provides different tools for querying the treebanks on-line.¹⁰ A relevant initiative of this kind is the Norwegian *Infrastructure for the Exploration of Syntax and Semantics* (INESS) (Rosén et al., 2012), which offers an open and easy-to-use platform for building, accessing, searching and visualizing treebanks through a web browser.¹¹

These collections and infrastructures enable to use and query various resources and tools from one common place on the web, but they do not provide a real interconnection between them, thus failing to achieve their interoperability. Instead, making linguistic resources interoperable requires that all types of annotation applied to a particular word/text get integrated into a common representation that enables access to the linguistic information conveyed in a linguistic resource or produced by an NLP tool (Chiarcos, 2012, p. 162). Particularly, by applying the principles of Linked Data to linguistic resources¹² “it is possible to follow links between existing resources to find other, related data and exploit network effects” (Chiarcos et al., 2013, p. iii).¹³ Despite their rich annotation (ranging from tokenization to syntactic analysis), treebanks alone cannot account for the linguistic complexity of the texts they include, which requires that information provided by different (and currently available) textual and lexical resources is interlinked and, thus, exploited to the best.

To this aim, the *LiLa: Linking Latin* project (2018-2023)¹⁴ was launched with the objective to interlink the wealth of linguistic resources and NLP tools for Latin developed thus far, in order to bridge the gap between raw language data, NLP and knowledge description (Declerck et al., 2012, p. 111). LiLa addresses this challenge by building a collection of several data sets described using the same vocabulary and linked together, namely a Linked (Open) Data Knowledge Base of the linguistic resources (and NLP tools) for Latin currently available from different providers under various licences.

After a brief description of the basic architecture of the LiLa Knowledge Base (Section 2), this paper focuses on the inclusion of three dependency treebanks for Latin into LiLa (namely, the IT-TB in two versions, PROIEL and the LLCT), presenting an example of a complex query crossing the treebanks and the other linguistic resources included so far in the Knowledge Base (Section 3).

2 The LiLa Knowledge Base

In order to achieve interoperability between linguistic resources and NLP tools, the LiLa Knowledge Base makes use of a set of Semantic Web and Linguistic Linked Open Data standards. These include ontologies to

⁵<http://www.thesaurus.badw.de/>

⁶<http://www.clarin.eu>

⁷<http://www.dariah.eu>

⁸<http://www.meta-share.org/>

⁹<http://www.eagle-network.eu>

¹⁰SETS treebank search (http://bionlp-www.utu.fi/dep_search); PML Tree Query (<http://lindat.mff.cuni.cz/services/pmltq/>); Kontext (<http://lindat.mff.cuni.cz/services/kontext/corpora/corplist>); Grew-match (<http://match.grew.fr/>).

¹¹<http://clarino.uib.no/iness/page>

¹²See Tim Berners-Lee’s note at <https://www.w3.org/DesignIssues/LinkedData.html>.

¹³The *Linguistic Linked Open Data cloud* <http://linguistic-lod.org/llod-cloud> is a good example of a set of interconnected linguistic resources.

¹⁴<https://lila-erc.eu/>

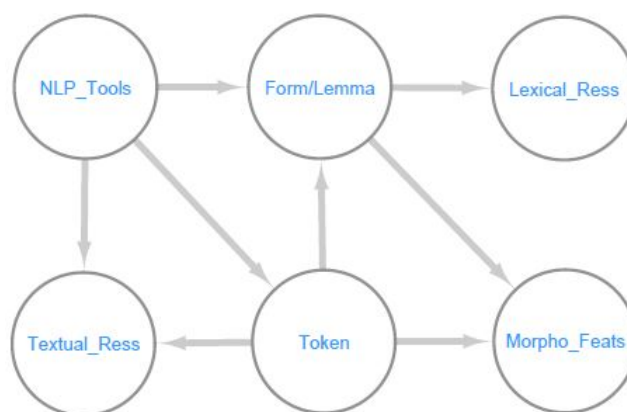


Figure 1: The basic architecture of the LiLa Knowledge Base.

describe linguistic annotation (OLiA (Chiarcos and Sukhareva, 2015)), corpus annotation (NIF (Hellmann et al., 2013), conll-rdf (Chiarcos and Fäth, 2017)) and lexical resources (Lemon (Buitelaar et al., 2011), Ontolex¹⁵). The Resource Description Framework (RDF) (Lassila et al., 1998) is used to encode graph-based data structures to represent linguistic annotations in terms of triples, made of a predicate connecting two nodes (a subject and its object). The SPARQL language is used to query the data recorded in the form of RDF triples (Prud’Hommeaux et al., 2008).

The LiLa Knowledge Base is highly lexically-based, striking a balance between feasibility and granularity: its basic assumption is that textual resources are made of (occurrences of) words, lexical resources describe properties of words, and NLP tools process words. Figure 1 presents the basic architecture of the LiLa Knowledge Base, showing its main components and their relations. The **Lemma** is the key node type in LiLa. A Lemma is an (inflected) **Form** conventionally chosen as the citation form of a lexical item. Lemmas occur in **Lexical Resources** as canonical forms of lexical entries. Forms, too, can occur in lexical resources, like in a lexicon containing all of the forms of a language, as for instance in Tombeur (1998). The occurrences of Forms in real texts are **Tokens**, which are provided by **Textual Resources**. Finally, **NLP tools** process either Forms regardless of their contextual use (e.g., a morphological analyzer), or Tokens (e.g., a PoS-tagger), or texts in Textual Resources (e.g., a tokenizer). Forms, Lemmas and Tokens can be assigned **Morphological Features**, like part of speech and gender.

Since lemmas serve as the optimal interface between lexical resources, (annotated) corpora and NLP tools, the core of the LiLa Knowledge Base is a collection of citation forms for Latin. Interoperability can be achieved by linking the entries in lexical resources and the corpus tokens pointing to the same lemma.¹⁶ The collection of citation forms of LiLa is built on top of the set of lemmas used by the morphological analyzer for Latin Lemlat (Passarotti et al., 2017).¹⁷ Lemlat relies on a lexical basis resulting from the collation of three Latin dictionaries (Georges and Georges, 1913 1918; Glare, 1982; Gradenwitz, 1904) for a total of 40,014 lexical entries and 43,432 lemmas, as more than one lemma can be included in one lexical entry. This lexical basis was recently further enlarged by adding the *Onomasticon* provided by the 5th edition of Forcellini dictionary (Budassi and Passarotti, 2016) and the entries from a large reference glossary for Medieval Latin, namely the *Glossarium Mediae et Infimae Latinitatis* (du Cange et al., 1883 1887; Cecchini et al., 2018), leading to a total of around 150,000 lemmas.

The linguistic resources currently linked in the LiLa Knowledge Base are stored in a triplestore using the Jena framework.¹⁸ The Fuseki component exposes the data as a SPARQL end-point accessible over HTTP. The current prototype of the LiLa RDF triplestore database connects the following resources for Latin: (a) the collection of lemmas provided by Lemlat, (b) the wFL lexicon, and (c) three treebanks (four by version):

¹⁵<https://www.w3.org/community/ontolex/>

¹⁶On the process of harmonization of the different lemmatization strategies for Latin in LiLa, see Mambrini and Passarotti (Forthcoming).

¹⁷<https://github.com/CIRCSE/LEMLAT3>

¹⁸A prototype of the LiLa triple store is available at <https://lila-erc.eu/data/>.

(c.1) PROIEL in its UD version (release 2.3), (c.2-3) the IT-TB in both its UD 2.3 and original version, and (c.4) a selection of 3,900 sentences (105,380 tokens) of the LLCT.

3 Interlinking and Querying Treebanks in LiLa

In this section, we discuss how we integrated the Latin treebanks into the LiLa Knowledge Base and how the linked data obtained by connecting the treebank tokens to the other resources support complex queries crossing through different linguistic resources.

3.1 Linked Treebanks

The Latin treebanks currently integrated into LiLa have been converted into RDF triples. As an example, Figure 2 represents a first result in the conversion and linking process. The figure shows a three-word sentence from the *Vulgata* (*Matt.* 6.10), taken from the UD 2.3 version of the PROIEL corpus: *veniat regnum tuum* (“thy kingdom come”). The UD 2.3 tree for this sentence is shown in Figure 3.¹⁹

Tokens and sentences are defined using the NIF vocabulary. In the current, preliminary stage of the Knowledge Base, some information on the tokens, such as the list of morphological features, is still registered as a simple string of text. For instance, in Figure 2 this is the case of the string “Case=Nom|Gender=Neut|Number=Sing”, which is linked to the PROIEL token with ID *s15924_2* (for the word *regnum* “kingdom”) via the relation `conll:FEAT`, linking the morphological features taken from files in the CoNLL-U format of UD.²⁰

Other types of tagging (such as syntactic dependencies, or sentence boundaries) are expressed by links between the nodes for tokens or sentences. For example, in Figure 2, this is represented by the linking between the token *s15924_2* (*regnum*) and the token *s15924_1* (*veniat* “come”) via the relation `conll:HEAD`, representing that in the sentence the word *veniat* is the head of the word *regnum*, as can be seen from the tree in Figure 3.

Finally, a third group of linguistic annotations, like the part of speech, directly relate tokens to concepts from an ontology of linguistic data (OLiA).²¹ In Figure 2, this is shown by the edge connecting the token *s15924_2* (*regnum*) to the concept node `olia:CommonNoun`.

Tokens are connected to the appropriate Lemma nodes recorded in the LiLa Knowledge Base. In Figure 2, for instance, the token *s15924_2* (*regnum*) is linked to lemma 34146, which has written representation *regnum*. Via this connection, it becomes possible to access all the other information that is also pointing to that lemma. In the figure, the lemma 34146 is connected to a node for a lexical base (1133), the same to which also lemmas *rex* “king” (34799) and *regno* “to rule, to be king” (34145) are attached. This means that lemmas *regno*, *regnum* and *rex* belong to the same “word formation family”, i.e. a set of lemmas sharing the same lexical base. The lemma *regnum* is also formed with the suffix “-n” (represented by the node `affix:111` in Figure 2), the same found in e.g. *fanum* “shrine” (not shown here for reasons of space). In the collection of citation forms included in LiLa, all the lemmas formed with the suffix “-n” are linked to `affix:111` via the relation `lemlat_base:hasSuffix`, thus allowing to retrieve them in the Knowledge Base. The information about lexical bases and affixes is available thanks to the connection of the WFL lexicon in LiLa.

3.2 Querying LiLa

In this section, we provide an example of the types of queries that the LiLa Knowledge Base can already support. As mentioned, one single query can extract data from all the multiple corpora and lexical resources linked to LiLa’s collection, and can also combine syntactic, lexical and morphological information beyond the type of annotation explicitly recorded in a single corpus.

¹⁹In Figure 3, each node apart from the root is assigned its part of speech and a dependency relation. In the tree, the `nsubj` relation is used for nominal subjects, while `nmod` for nominal modifiers. The full list of dependency relations used in UD v2 is available at <https://universaldependencies.org/u/dep/index.html>.

²⁰On the CoNLL-U format used in the UD treebanks see <https://universaldependencies.org/format.html>.

²¹A shallow conversion from the CoNLL-U format to RDF was obtained with the help of `conll-rdf`. The application also allows to design custom SPARQL Update queries to link the RDF representation of the corpus to other resources.

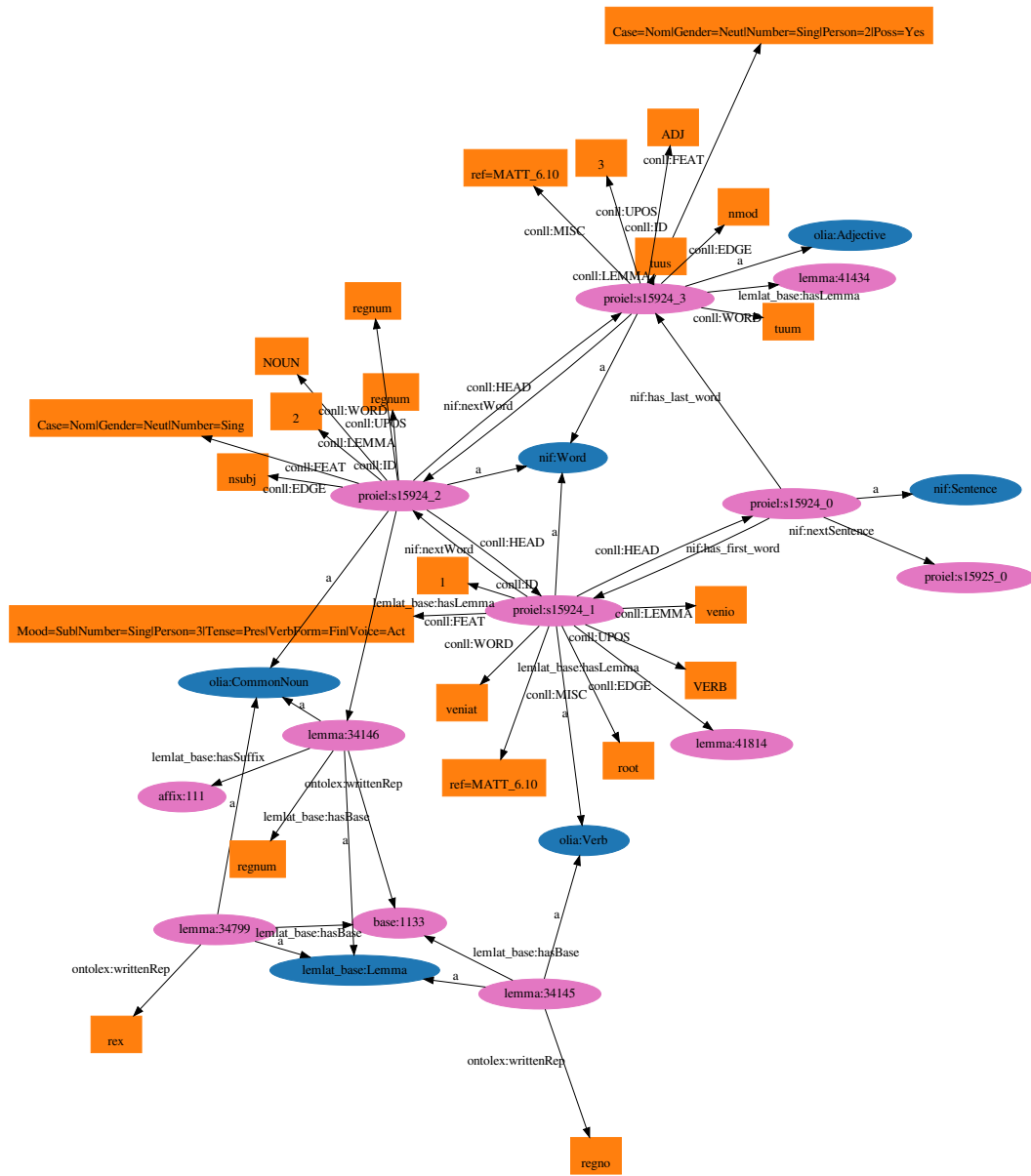


Figure 2: A sentence from PROIEL as RDF triples in the LiLa Knowledge Base.

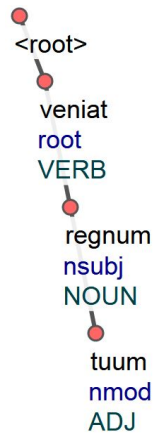


Figure 3: The UD 2.3 tree of *veniat regnum tuum* from PROIEL.

Consider, for instance, the case of a researcher interested in the relation between the syntactic role of subject and the semantic role of agent in Latin. One possible approach to study the question would be to start by collecting and analyzing the sentences where nouns formed with a typical morpheme for agent nouns like “-(t)or” (common to several Indo-European languages) are attested as subject of an active verb.

Though the number of linguistic resources currently interlinked in LiLa is still small, it is already possible to design a single SPARQL query to extract this information from our RDF versions of PROIEL, IT-TB (UD version) and LLCT. In what follows, we illustrate the results of a query that asks for an active (or dependent) verb governing a noun with the syntactic relation of subject in the three treebanks. By leveraging the connection between lemmas and the affixes in WFL, we add the additional constraint that the noun must be formed with the suffix “-(t)or”. This information, which is not encoded into the original treebanks, is now accessible thanks to the architecture based on Linked Open Data that LiLa adopts.

The query allows us to extract 143 passages, with 80 different verbs and 58 agent nouns. One sample of the results, a sentence from Cicero’s *Letters to Atticus* (4.4a.2) retrieved from PROIEL, is reported in Example (1).

- (1) **gladiatores** audio **pugnare** mirifice.
‘I hear that your gladiators fight superbly.’

The subject-verb bigrams resulting from the query highlight interest lexical aspects in the language of the three corpora. As it is to be expected from the documentary nature of the texts provided by the LLCT treebank, the 10 occurrences found in this corpus all involve legal actors and events: the most frequent subject (4 occurrences) is *rector*, the priest responsible for a rural church. The other actors are: *dispensator* “treasurer”, *fideiussor* “bail”, *genitor* “parent” and *imperator* “emperor”.

In the ITTB, on the other hand, the most frequent couplet is the one formed by the noun *commentator* “interpreter” and the verb *dico* “to say” (21 cases), where the assertions of a scholar are reported and discussed. Indeed, the verbs pointing to intellectual activities of scholars dominate in the results from the corpus of Thomas Aquinas: in addition to the most frequent *dico* (22), other intellectual verbs include *respondeo* “to reply” (3 instances), *finjo* “to imagine” (2), and *intendo* “to mean” (2).

Finally, PROIEL, which is more balanced between different genres, offers a more varied set of subject-verb couplets in its 57 results. As in Example (1), where the noun *gladiator* “gladiator” is coupled with the verb *pugnare* “to fight”, we find several nouns and verbs from everyday life, or from the domain of the professions and human activities. Thus, for instance, we find 4 cases of *fossor* “digger, ditcher” joined with verbs like *includo* “to shut in” and *incumbo* “to press upon”, or 6 cases of *pastor* “herdsman, shepherd” with verbs like *fugio* “to flee” and *secludo* “to shut off”.

4 Conclusions and Future Work

In this paper, we have described how we interlinked three dependency treebanks for Latin (one available in two versions) into a Knowledge Base of linguistic resources based on Linked Open Data practices and standards. Linking resources of different kind (such as corpora and lexica) makes it possible to exploit their potential to the best. Indeed, single resources tend to focus on a limited set of linguistic features (e.g. morphology and syntax for treebanks), which are in most cases insufficient to provide a full analysis of the textual or lexical data. Making interoperable the still scattered and unconnected resources that are currently available for Latin (as well as for many other languages) is a way to approach the data from the various layers of annotation that such resources provide.

Our work of interlinking the linguistic resources for Latin has just begun. In the near future, we plan to integrate into the LiLa Knowledge Base two other lexical resources, namely an etymological dictionary (de Vaan, 2008) and the Latin WordNet. Interlinking these resources with the textual occurrences of their lemmas (enriched with syntactic annotation in treebanks) will enable the users of LiLa to run complex queries crossing different kinds of linguistic features. Given that the set of interlinked resources will grow in the coming years, the chain of connection can be continued indefinitely; as long as new lexical resources are connected to the Knowledge Base, all the connections from any corpus token to their nodes will become explorable in the network.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme - Grant Agreement No 769994.

References

- David Bamman and Gregory Crane. 2006. The design and use of a Latin dependency treebank. In *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT 2006)*, pages 67–78, Prague, Czech Republic. Univerzita Karlova.
- David Bamman, Marco Passarotti, Gregory Crane, and Savina Raynaud. 2007. Guidelines for the syntactic annotation of latin treebanks. *Tufts University Digital Library*.
- Marco Budassi and Marco Passarotti. 2016. Nomen omen. Enhancing the Latin morphological analyser Lemlat with an onomasticon. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 90–94, Berlin, Germany. Association for Computational Linguistics.
- Paul Buitelaar, Philipp Cimiano, John McCrae, Elena Montiel-Ponsoda, and Thierry Declerck. 2011. Ontology lexicalisation: The lemon perspective. In *Proceedings of the Workshops. 9th International Conference on Terminology and Artificial Intelligence*, pages 33–36.
- Flavio Cecchini, Marco Passarotti, Paolo Ruffolo, Marinella Testori, Lia Draetta, Martina Fieromonte, Annarita Liano, Costanza Marini, and Giovanni Piantanida. 2018. Enhancing the latin morphological analyser lemlat with a medieval latin glossary. In *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018). 10-12 December 2018, Torino*, pages 87–92.
- Christian Chiarcos and Christian Fäth. 2017. CoNLL-RDF: Linked Corpora Done in an NLP-Friendly Way. In Jorge Gracia, Francis Bond, John P. McCrae, Paul Buitelaar, Christian Chiarcos, and Sebastian Hellmann, editors, *Language, Data, and Knowledge*, pages 74–88, Cham. Springer International Publishing.
- Christian Chiarcos and Maria Sukhareva. 2015. OLiA - Ontologies of Linguistic Annotation. *Semantic Web Journal*, 6(4):379–386.
- Christian Chiarcos, Philipp Cimiano, Thierry Declerck, and John P McCrae. 2013. Linguistic linked open data (llod). introduction and overview. In *Proceedings of the 2nd Workshop on Linked Data in Linguistics (LDL-2013): Representing and linking lexicons, terminologies and other language data*, pages i–xi.
- Christian Chiarcos. 2012. Interoperability of corpora and annotations. In *Linked Data in Linguistics*, pages 161–179. Springer.
- Michiel de Vaan. 2008. *Etymological Dictionary of Latin and the other Italic Languages*. Leiden & Boston: Brill.
- Thierry Declerck, Piroska Lendvai, Karlheinz Mörtz, Gerhard Budin, and Tamás Váradi. 2012. Towards linked language data for digital humanities. In *Linked Data in Linguistics*, pages 109–116. Springer.
- Charles du Fresne du Cange, Bénédicins de Saint-Maur, Pierre Carpentier, Louis Henschel, and Léopold Favre. 1883–1887. *Glossarium mediae et infimae latinitatis*. Niort, France.
- Karl Ernst Georges and Heinrich Georges. 1913–1918. *Ausführliches lateinisch-deutsches Handwörterbuch*. Hahn, Hannover, Germany.
- Peter GW Glare. 1982. *Oxford Latin dictionary*. Clarendon Press. Oxford University Press, Oxford, UK.
- Otto Gradenwitz. 1904. *Laterculi Vocum Latinarum: voces Latinas et a fronte et a tergo ordinandas*. Hirzel, Leipzig, Germany.
- Dag TT Haug and Marius Jøhndal. 2008. Creating a parallel treebank of the old Indo-European Bible translations. In *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, pages 27–34, Marrakesh, Morocco. European Language Resources Association (ELRA).
- Dag Haug. 2010. Proiel guidelines for annotation. Retrieved April, 23:2013.
- Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. 2013. Integrating NLP using Linked Data. In *12th International Semantic Web Conference, Sydney, Australia, October 21-25, 2013*.

- Nancy Ide and James Pustejovsky. 2010. What does interoperability mean, anyway. *Toward an Operational*.
- Timo Korkiakangas and Marco Passarotti. 2011. Challenges in annotating medieval latin charters. *Journal for Language Technology and Computational Linguistics*, 26(2):103–114.
- Ora Lassila, Ralph R. Swick, World Wide, and Web Consortium. 1998. Resource description framework (rdf) model and syntax specification.
- Eleonora Litta, Marco Passarotti, and Chris Culy. 2016. *Formatio formosa est*. building a word formation lexicon for latin. In *Proceedings of the third italian conference on computational linguistics (clic-it 2016)*, pages 185–189.
- Francesco Mambrini and Marco Passarotti. Forthcoming. Harmonizing different lemmatization strategies for building a knowledge base of linguistic resources for latin. In *Proceedings of the 13th Linguistic Annotation Workshop (LAW XIII)*, Florence, Italy.
- Stefano Minozzi. 2010. The latin wordnet project. In *Latin Linguistics Today. Latin Linguistics Today. Akten des 15. Internationalen Kolloquiums zur Lateinischen Linguistik*, pages 707–716.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher 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)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).
- Marco Passarotti, Marco Budassi, Eleonora Litta, and Paolo Ruffolo. 2017. The Lemlat 3.0 Package for Morphological Analysis of Latin. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, volume 133, pages 24–31. Linköping University Electronic Press.
- Marco Passarotti. 2011. Language resources. The state of the art of Latin and the *Index Thomisticus* treebank project. In Marie-Sol Ortola, editor, *Corpus anciens et Bases de données*, number 2 in ALIENTO. Échanges sapientiels en Méditerranée, pages 301–320, Nancy, France. Presses universitaires de Nancy.
- Edoardo Maria Ponti and Marco Passarotti. 2016. *Differentia compositionem facit*. A slower-paced and reliable parser for Latin. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 683–688, Portorož, Slovenia. European Language Resources Association (ELRA).
- Eric Prud’Hommeaux, Andy Seaborne, et al. 2008. Sparql query language for rdf. w3c. *Internet: <https://www.w3.org/TR/rdf-sparql-query/>*[Accessed on February 27th, 2019].
- Victoria Rosén, Koenraad De Smedt, Paul Meurer, and Helge Dyvik. 2012. An open infrastructure for advanced treebanking. In *META-RESEARCH Workshop on Advanced Treebanking at LREC2012*, pages 22–29. Hajič, Jan.
- Paul Tombeur. 1998. *Thesaurus formarum totius Latinitatis: a Plauto usque ad saeculum XXum*. Turnhout: Brepols.

Challenges of Annotating a Code-Switching Treebank

Özlem Çetinoğlu

IMS

University of Stuttgart

Germany

ozlem@ims.uni-stuttgart.de

Çağrı Çöltekin

Department of Linguistics

University of Tübingen

Germany

ccoltekin@sfs.uni-tuebingen.de

Abstract

This paper presents challenges and observations on creating a code-switching treebank based on ongoing annotation efforts of a Turkish–German spoken corpus following the Universal Dependencies annotation scheme. We present and discuss a number of issues that arise because of the need for consistent multilingual annotation within a single treebank, as well as the informal language which is where code-switching is observed most. Besides proposing solutions to these issues, our aim in this paper is to stimulate discussion and facilitate consistency over upcoming code-switching annotation projects.

1 Introduction

Code-switching (CS) is the process of mixing more than one language in written or spoken communication (Myers-Scotton, 1993; Poplack, 2001; Toribio and Bullock, 2012). It is a phenomenon commonly observed in multilingual societies (Auer and Wei, 2007), mainly in informal settings such as social media and spoken communication. For instance, (1) shows a sentence from a dialogue, that mixes Turkish and German (in bold). The speaker starts with Turkish, switches to German, back to Turkish, and ends the sentence with a mixed word where the German noun *Gastfamilie* ‘host family’ is inflected with the Turkish locative suffix *-de*.

- (1) Eh orada iki **Wochen** kaldım **ehm Gastfamiliede ehm** .
Eh there two week.Pl stay.Past.1sg ehm guest family.Loc ehm .
‘I stayed there two weeks, in a host family.’

The sentence is relatively simple and the overall meaning is derivable from the individual words. Yet, its syntax is not standard. The main predicate *kaldım* ‘I stayed’ is in Turkish and the whole sentence seemingly follows the Turkish syntax, except the noun phrase *iki Wochen* ‘two weeks’. Nouns modified by numbers are in singular in Turkish, but *Wochen* is in plural. The construction is more complex than using the German equivalent of ‘week’ in a Turkish phrase. It seems, the speaker inherently switches to the German syntax as well, where the noun should be plural, when switching to German on the surface.

Such CS-specific constructions vary from non-canonical morphological marking to creating new syntactic representations, to applying a linguistic phenomenon of one language to the other. They make structural analysis of code-switching linguistically interesting and computationally challenging. Several approaches tackle these challenges by utilising labelled and unlabelled monolingual and parallel data, e.g. by creating artificial CS data and using them in training models for processing CS (Pratapa et al., 2018; Zhang et al., 2018). However to be able to capture unique cases like the singular-to-plural mapping for ‘week’ in (1), those models need to see such instances. Thus, to observe the characteristics of CS and address them with data-driven tools, we are in the process of annotating Turkish–German transcriptions with part-of-speech, morphology, and dependency layers.

We have chosen Universal Dependencies (UD) (Nivre et al., 2016) as our annotation scheme. The UD project aims to define morphosyntactic annotation guidelines that are consistent across languages. Its unified tag sets and annotation standards facilitate the annotation of multiple languages within a single treebank. Furthermore, annotations parallel to monolingual resources are useful for making use of these resources, e.g., for transfer learning (Bhat et al., 2018).

Despite clear advantages of the UD framework for annotating CS treebanks, the annotation of multiple languages in a single treebank needs additional considerations that have not been studied before. Although there has been a few UD treebanks with code-switching (Bhat et al., 2018; Partanen et al., 2018), the papers describing these treebanks do not document or discuss the code-switching aspects of the annotation process.

In this paper we address this gap and outline some of the challenges and interesting phenomena that surface during the annotation of a Turkish–German code-switching treebank. Our contributions are in two levels. The observations on code-switching, independent of the annotation scheme, help in understanding in what forms it occurs. The annotation solutions we propose explore how to handle CS within the UD framework. Working with spoken data brings another aspect and opens also speech annotation under UD to discussion.

2 Related Work

Many well-known linguistic theories on CS syntax, e.g. Free Morpheme and Equivalence Constraints (Poplack, 1980), Closed-class Constraint (Joshi, 1982), Matrix Language Frame (Myers-Scotton, 1993), Functional Head Constraint (Belazi et al., 1994) define their formalism and constraints on constituency structures. Eppler (2005) argues that these constraints are too restrictive from a data-driven perspective and favours Word Grammar (Hudson, 1990), a dependency-based formalism, where the scope of the constraints is head-dependent pairs. Her annotations on German–English transcriptions and the Chinese–English treebank (Wang and Liu, 2013), which also follows Word Grammar, are the only CS dependency treebanks that do not follow UD to the best of our knowledge.

The starting point for our work is the monolingual UD treebanks of both languages in our study. The recent 2.4 release of UD includes three Turkish and four German treebanks. Turkish treebanks include IMST-UD (Sulubacak et al., 2016b), which is semi-automatically converted from the IMST treebank (Sulubacak et al., 2016a) which, in turn, is a re-annotation of the METU-Sabancı treebank (Oflaz et al., 2003). Turkish GB is a manually annotated treebank consisting of grammar book examples (Çöltekin, 2015). There are PUD treebanks consisting of parallel (translated) sentences for both languages. The PUD treebanks were automatically converted from another dependency scheme for the CoNLL 2017 multi-lingual parsing shared task (Zeman et al., 2017). The first German UD treebank is the GSD treebank (McDonald et al., 2013), which is also automatically converted from a different dependency formalism. There are also two new additions to German treebanks; HDT, a conversion of Hamburg Dependency Treebank (Foth et al., 2014; Hennig and Köhn, 2017), and LIT, a treebank of German literary history. Most of our annotation decisions and the discussions below are based on the version 2.3 of the UD treebanks, particularly Turkish IMST and German GSD. There are, however, inconsistencies across languages, and across treebanks of the same language. For most annotation decisions, we follow the annotations in the monolingual treebanks as much of possible. In case of inconsistencies across treebanks, our policy is to choose the alternative closest to the general UD guidelines, so as to ensure cross-lingual consistency within our multilingual treebank.

None of the treebanks noted above include spoken language, let alone code-switching. Quite a few UD treebanks, on the other hand, contain spoken language partially (Danish DDT, Greek GDT, Latvian LVTB, Persian Seraji, Polish LFG, Swedish LinES) or fully (Cantonese HK, Chinese HK, French Spoken, Naija NSC, Norwegian NynorskLIA, Slovenian SST). These treebanks have extended the UD dependency relations with subtypes in addition to using the existing ones to cover linguistic phenomena mainly observed in speech. For example, Slovenian SST (Dobrovoljc and Nivre, 2016) annotates correcting disfluencies either with `reparandum` or `parataxis:restart`. Another `parataxis` subtype, `parataxis:discourse` is defined to cover sentential parentheticals with fixed semantics that serve as discourse elements (e.g., *you know*). French Spoken (Gerdes and Kahane, 2017) and Naija NSC (Courtin et al., 2018) employ the same tag too. They define a separate tag `parataxis:dislocated` for clauses that precede the sentence they are dislocated from. The other relation that is commonly extended is `discourse`. Slovenian SST separates filler sounds from other discourse elements and assigns them `discourse:filler`. Norwegian NynorskLIA (Øvrelid and Hohle, 2016) follows the same approach.

Cantonese HK and Chinese HK (Leung et al., 2016) define `discourse:sp` for sentence particles common in spoken language. So far we are more conservative in extending relations with subtypes and have introduced one that is described in Section 3.2.

The Hindi-English UD treebank (Bhat et al., 2018) annotates the mixed language of social media and has no extension to UD dependencies. The major annotation augmentation is the language IDs assigned to each token. Komi-Zyrian IKDP (Partanen et al., 2018) consists of spoken language, and some utterances include Russian phrases. In those utterances mixed and Russian tokens are marked with respective language IDs, and the Russian syntax is applied. However, the authors do not claim any consistency with the annotations of the monolingual Russian UD treebank. Similar to these treebanks, we also assign a language ID to each token following the tag set in Çetinoğlu (2016). Many other treebanks include words or phrases from a foreign language. Most of them mark foreign tokens with `Foreign=Yes`, and annotate the internal structure of foreign phrases with `flat` relations. However, a few treebanks, e.g., Irish IDT (Lynn and Foster, 2016), annotate foreign tokens according to their respective language.

3 Annotations

Any annotation project is bound to make non-trivial choices (Gerdes and Kahane, 2016). Most non-trivial choices for a code-switching treebank comes either because of the multilingual nature of the resource, or, as noted earlier, the fact that code-switching is prevalent in informal language, and annotation of informal or spoken language has been more challenging than more standard/written language. Most of the problems related to multilingual nature of the data stem from different annotations choices established for individual languages. Although one of the main motivations behind the UD project is multilingual consistency across treebanks, multilingualism within a treebank has not been one of the motivations for UD. Below we focus on issues that arise due to multilingual nature of the treebank, but also noting some of the issues that are due to the informal and spoken language.

3.1 Annotation Differences in Individual Languages

To be able to benefit maximally from monolingual treebanks, one of the principles we follow is to annotate the tokens that belong to each language following the annotation standards in the monolingual treebank(s) of the corresponding language. In many cases this produces a workable solution in a multilingual treebank. In other cases, however, the interaction of tokens within a sentence results in conflicts. In this section we provide a few examples of both cases.

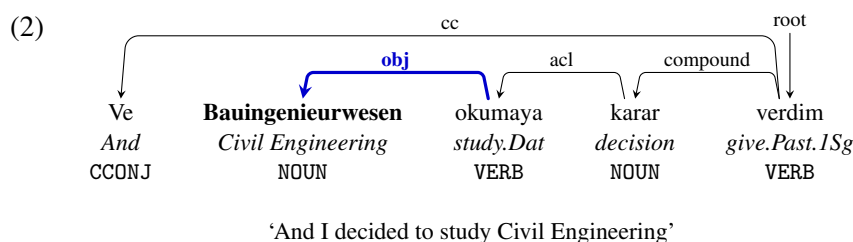
Titles A relatively simple difference between existing monolingual German and Turkish treebanks is the annotation of titles, e.g., as in *President Obama*. The UD guidelines prescribe the use of `flat` relation here. However, the different treebanks follow slightly different practices.¹ German treebanks seem to annotate names using `appos` relation. In Turkish treebanks, similar to a few other treebanks in the UD distribution, the `nmod` relation is used. Although this is a relatively trivial issue, it demonstrates the trade-offs of the annotation choices. On the one hand, choosing one of three relations and applying to both languages would cause inconsistency with the (larger) monolingual treebanks and tools based on these treebanks. On the other hand, following the conventions of both languages causes inconsistency within the multilingual treebank, potentially confusing users querying the treebank, or automatic tools that are trained on it.

Copula Another similar issue is the annotation of different sort of copula in German. One of the principles of Universal Dependencies is the primacy of the content words. For copular constructions, this means marking the copula as the dependent rather than the head. Since a copula is rarely used in Turkish, the Turkish treebanks naturally follow this for all types of copular constructions. On the other hand, the German GSD and PUD treebanks seem to make distinction where some uses of copula *sein* is annotated as main verb. For example, these treebanks suggest that copula *ist* in *Die Frau ist Ärztin* ‘the woman is a doctor’ should be annotated using `cop` (with head *Ärztin*), while in *Der Vortrag ist in dem*

¹See <https://universaldependencies.org/workgroups/mwe.html> for a discussion.

großen Saal ‘The lecture is in the great hall’, it should be marked as the main verb.²

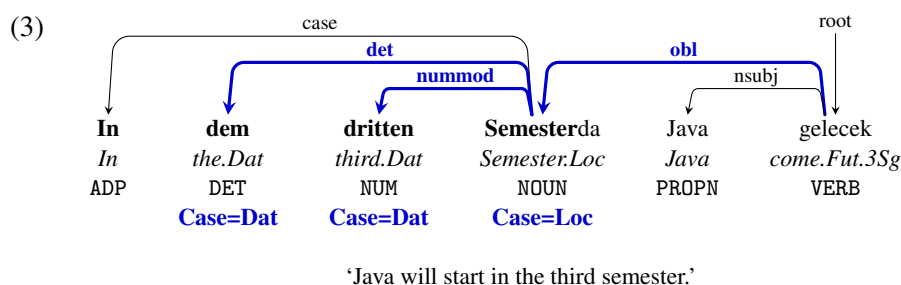
Case A particular issue in Turkish–German CS occurs due to different approaches in annotating morphology. Traditionally, morphological annotations in German treebanks are fully disambiguated based on syntax (and possibly larger context) of the sentence. Although not clear-cut, Turkish treebanks annotate only the morphological features that can be inferred from the word form alone. For example, without context, German nouns belonging to some gender classes are ambiguous with respect to their cases. The word (*das*) *Kind* ‘(the) child’ would be annotated with *Case=Nom* if it is the subject, and with *Case=Acc* if it is the object of the sentence. A similar ambiguity also exists in Turkish. The word *çocuk* ‘child’ may be either the subject or indefinite object of a sentence. However, in both cases it is tagged with *Case=Nom*. The tag *Case=Acc* is only used for definite objects where there is an overt morphological marking for case.



(2) presents a sentence involving a German word that functions as an object of a Turkish predicate. According to German annotation standards, the word should be tagged as *Case=Acc*. However, there is no overt case marker,³ thus the tag should be *Case=Nom* according to Turkish annotation standards. The principle of following the annotation scheme of the token’s language does not work well here, causing the loss of the distinction between definite and indefinite objects in Turkish. In such cases, we chose the language of the head as reference.

3.2 CS-specific Issues

Double case marking Annotating case marking can get more complicated when it is overt in both languages. In (3), the article *dem* ‘the’ and the number *dritten* ‘third’ carry the dative case marking to indicate the static meaning. The noun *Semester* normally does not carry an explicit marker and the German phrase *in dem dritten Semester* ‘in the third semester’ would be completely grammatical. Thus the token *Semester* would normally have the tag *Case=Dat* in its morphological annotation in agreement with its modifiers. However, the speaker has chosen to mark the static meaning *also* in Turkish and following the Turkish grammar rules, there is a locative case marker *-da* attached to the noun, which entails a *Case=Loc* tag in its morphological representation.



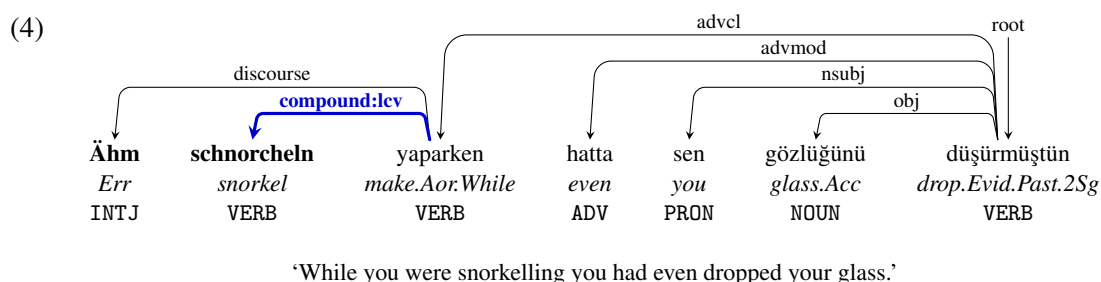
The conflict between case markers does not have a perfect solution within the current UD representation. If we choose *Case=Dat* to follow the German rules, the surface form *-da* would not match the morphological tag, furthermore it would change the semantics of the word, as the dative case represents

²Our design decisions are mainly based on treebanks released in UD version 2.3. As of version 2.4, HDT and LIT treebanks are released for German. While LIT follows GSD and PUD in copula annotation, HDT mark them with *cop*, in accordance with the general and Turkish guidelines. Thus the German copular representation is subject to change.

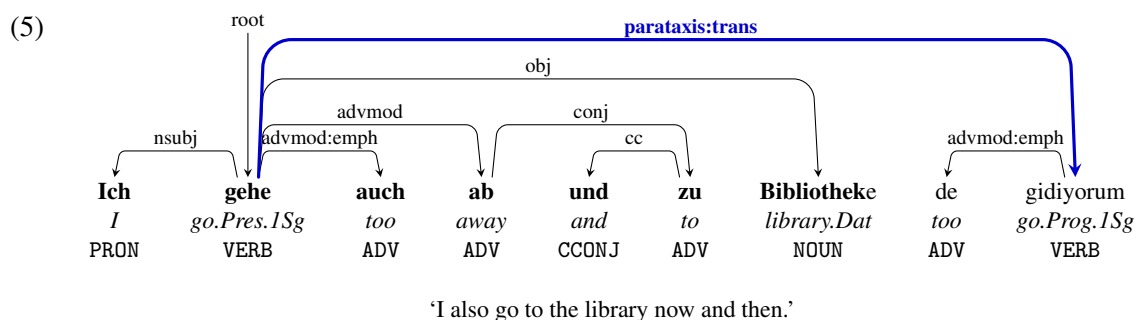
³Note that *Bauingenieurweseni* – the version with the Turkish accusative case marker – would also be grammatical.

motion towards something in Turkish. Thus, we choose the Case=Loc tag at the expense of losing the agreement between the determiner and number, and the noun.⁴

Bilingual light verb constructions The use of CS creates new constructions too. One quite common new construction is the use of German verbs followed by a Turkish light verb *etmek* ‘do’ or *yapmak* ‘make’, which is also observed in Turkish–German tweets (Çetinoğlu, 2016) as well as Turkish–Dutch (Backus, 2009). The German verb is in infinitive form and the Turkish light verb takes inflectional and derivational suffixes. The core semantics of the construction comes from the German verb. These constructions are similar to noun-light verb constructions common in Turkish (e.g. *yardım etmek* lit. ‘help do’ – ‘to help’). In the Turkish UD, noun-verb constructions are labelled with the compound:1vc relation where 1vc denotes light verb constructions. We adopt the same label for German-Turkish constructions. (4) demonstrates a sentence where the German verb *schnorcheln* ‘snorkel’ is coupled with the Turkish light verb *yap* ‘make’, that undergoes derivation with the suffix *-ken* ‘While’. The combined meaning of the compound is ‘while snorkelling’.



Translation pairs Another CS-specific language use we have observed is uttering a word, phrase or clause in one language and repeating it as a translation in the other language. (5) shows such an example where German *gehe auch* ‘I go too’ is repeated again as Turkish *de gidiyorum*. Since there are no relations in UD that would capture this phenomenon, we extend the relation *parataxis* by introducing a *trans* subtype. The relation connects the head of the second constituent to the head of the first constituent as a dependent.



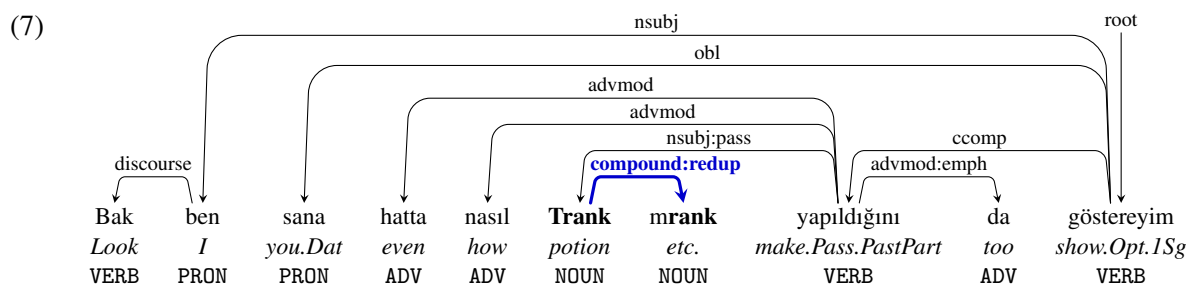
Bilingual m-reduplication In Turkish, it is possible to generalise the meaning of a word by so-called *m*-reduplication (Göksel and Kerslake, 2005). To realise *m*-reduplication, the first word is reduplicated, and an *m* prefixes the duplicate if the word starts with a vowel, or the first character of the duplicate is replaced with an *m* if it is a consonant as in (6).

- (6) Çay may içer misin?
Tea etc. drink.Aor Ques.2Sg
 ‘Would you like to drink tea and the like?’

While this is a Turkish-specific phenomenon, bilinguals also apply it to other languages. In (7) we see that the German word *Trank* ‘potion’ undergoes *m*-reduplication. This is not only a new lexical

⁴Another possibility is indicating both cases with notation Case=Dat, Loc. This is used when the word may have one of the values, but it cannot be decided from the available context. In this particular case, however, there is no ambiguity. Both case values are correct depending on the language.

alternation in German, its syntactic representation is new to German UD as well. *m*-reduplications are represented as `compound:redup` in the Turkish UD treebanks; we apply it also to German in this case.

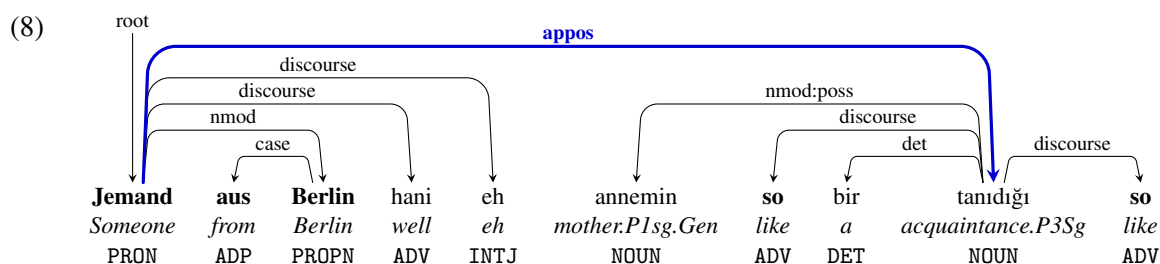


‘Look, let me even show you also how potion et cetera is made.’

3.3 Issues Related to Spoken Language

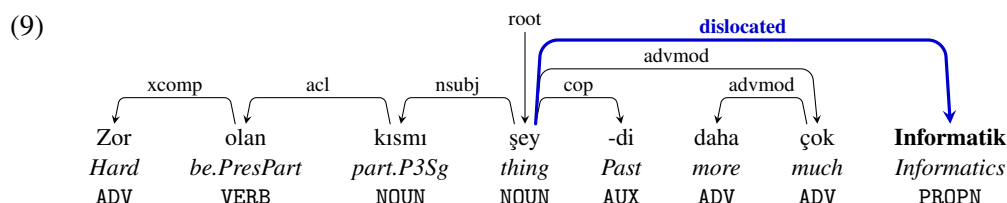
We also observe some linguistic phenomena more frequently than corresponding monolingual treebanks due to the medium we collect the data. Spoken language contains many disfluencies, repetitions, run-on sentences, and uncommon word order. Since these phenomena are orthogonal to mixing languages, their dependencies can cross language boundaries within a sentence. We exemplify two of the commonly observed cases.

Appositions In appositions two consecutive noun phrases define the same referent in different ways. In our corpus these two noun phrases could as well be in different languages. In (8) the speaker mentions ‘someone from Berlin’ in Turkish then refers to the same person with additional information ‘an acquaintance of my mother’ in German. Following the UD guidelines, the head of the second phrase is dependent on the head of the first phrase with the relation `appos`.



‘Someone from Berlin, well, an acquaintance of my mother.’

Dislocation In spoken Turkish it is quite common to replace a word or phrase that does not come to mind immediately or inappropriate to say with the word *şey* ‘thing’. While it is a noun itself, it can also replace verbs or clauses when combined with the light verb *etmek* ‘do’. The CS corpus we are collecting has many instances of such use, (9) demonstrates one case.

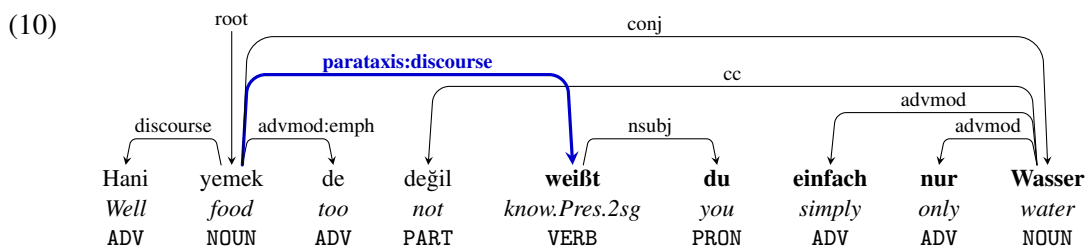


‘The hard part of it was mostly this thing, Informatics.’

The speaker first uses *şey* as the nominal predicate of the copular sentence. This way the sentence is grammatically complete with the placeholder *şey* until the last word. Once the word *İnformatik* ‘Informatics’ is uttered, it does not have a role in the sentence other than clarifying *şey*. UD employs the

dislocated tag for these relations. By definition the dislocated item is attached to the head of the placeholder. Here, the head is the placeholder itself, thus *Informatik* is dependent on *şey*.

Clausal discourse elements Spoken language contains many clauses with fixed semantics that function as discourse markers such as *you know, say, I think*. We observe similar cases in our corpus too; most frequent examples include German *weißt du* ‘you know’, *ich glaube* ‘I think’, and Turkish *bak* ‘look’. The UD policy for such cases is connecting them to the main clause with a parataxis tag. Some of the UD spoken treebanks (Dobrovoljc and Nivre, 2016; Gerdes and Kahane, 2017; Courtin et al., 2018) keep the discourse information via the subtype `parataxis:discourse`. We follow their approach and employ the same tag as exemplified in (10) with *weißt du* ‘you know’.



‘Well, it is not food, you know, just water.’

4 Conclusions

In this paper we present our experience with an ongoing treebank creation project of a Turkish-German code-switching corpus. In annotations, we follow the general UD guidelines and, Turkish and German UD treebanks as much as differences in individual languages allow. When we encounter new monolingual or bilingual syntactic constructions we apply existing relations to these new conditions; and if not sufficient, we introduce a subtype. Due to annotating spoken data, our sentences contain dependencies that are rare or nonexistent in monolingual Turkish and German treebanks. For those cases also, we follow general UD guidelines and other spoken UD treebanks.

Our observations so far suggest that interesting phenomena we come across and challenges they bring can only increase as we continue to collect and annotate more data. For some of the challenges we propose well-fitting solutions. For others, we take advantage of reporting work in progress and open our decisions up for discussion. Thus we see this paper as an opportunity to share idiosyncrasies of code-switching with any researcher who is interested in CS in particular, or in non-canonical language in general; and to exchange annotation ideas with the UD community.

Acknowledgements

We thank Cansu Turgut and Sevde Ceylan for data collection and annotation, and for discussions on the semantics of examples. We also thank the reviewers for their helpful comments. The first author is funded by DFG via Project CE 326/1-1 “Computational Structural Analysis of German-Turkish Code-Switching”.

References

- Peter Auer and Li Wei. 2007. *Handbook of multilingualism and multilingual communication*, volume 5. Walter de Gruyter.
- A. Backus, 2009. *Codeswitching as one piece of the puzzle of language change: The case of Turkish yapmak*, pages 307–336. Number 41 in *Studies in Bilingualism*. John Benjamins. Pagination: 20.
- Hedi M Belazi, Edward J Rubin, and Almeida Jacqueline Toribio. 1994. Code switching and x-bar theory: The functional head constraint. *Linguistic inquiry*, pages 221–237.

- Irshad Bhat, Riyaz A. Bhat, Manish Shrivastava, and Dipti Sharma. 2018. Universal Dependency parsing for Hindi-English code-switching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 987–998, New Orleans, Louisiana, June. Association for Computational Linguistics.
- Özlem Çetinoğlu. 2016. A Turkish-German code-switching corpus. In *The 10th International Conference on Language Resources and Evaluation (LREC-16)*, Portorož, Slovenia.
- 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.
- Marine Courtin, Bernard Caron, Kim Gerdes, and Sylvain Kahane. 2018. Establishing a language by annotating a corpus: the case of Naija, a post-creole spoken in Nigeria. In *Proceedings of the Workshop on Annotation in Digital Humanities*, pages 7–11, August.
- Kaja Dobrovoljc and Joakim Nivre. 2016. The Universal Dependencies treebank of spoken Slovenian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, may.
- Eva Maria Eppler. 2005. *The syntax of German-English code-switching*. Ph.D. thesis, University of London.
- Kilian A. Foth, Arne Köhn, Niels Beuck, and Wolfgang Menzel. 2014. Because size does matter: The Hamburg dependency treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2326–2333, Reykjavik, Iceland, May. European Language Resources Association (ELRA).
- Kim Gerdes and Sylvain Kahane. 2016. Dependency annotation choices: Assessing theoretical and practical issues of Universal Dependencies. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 131–140, Berlin, Germany, August. Association for Computational Linguistics.
- 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. In *Atelier sur les corpus annotés du français (ACor4French)*.
- Aslı Göksel and Celia Kerslake. 2005. *Turkish: A comprehensive grammar*. Routledge.
- 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.
- Richard A. Hudson. 1990. *English Word Grammar*. Oxford:Blackwell.
- Aravind K Joshi. 1982. Processing of sentences with intra-sentential code-switching. In *Proceedings of the 9th Conference on Computational Linguistics-Volume 1*, pages 145–150. Academia Praha.
- Herman Leung, Rafaël Poiret, Tak-sum Wong, Xinying Chen, Kim Gerdes, and John Lee. 2016. Developing Universal Dependencies for Mandarin Chinese. In *Proceedings of the 12th Workshop on Asian Language Resources (ALR12)*, pages 20–29, Osaka, Japan, December.
- Teresa Lynn and Jennifer Foster. 2016. Universal Dependencies for Irish. In *Celtic Language Technology Workshop*, pages 79–92.
- Ryan McDonald, Joakim Nivre, Yvonne Quirnbach-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.
- Carol Myers-Scotton. 1993. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher 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'16)*, pages 1659–1666.
- Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. 2003. Building a Turkish treebank. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, chapter 15, pages 261–277. Springer.

- 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, Brussels, Belgium, November. Association for Computational Linguistics.
- Shana Poplack. 1980. Sometimes I'll start a sentence in Spanish y termino en Espanol: toward a typology of code-switching. *Linguistics*, 18(7-8):581–618.
- Shana Poplack. 2001. Code-switching (linguistic). *International encyclopedia of the social and behavioral sciences*, pages 2062–2065.
- Adithya Pratapa, Monojit Choudhury, and Sunayana Sitaram. 2018. Word embeddings for code-mixed language processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3067–3072, Brussels, Belgium, October-November. Association for Computational Linguistics.
- Umut Sulubacak, Gülşen Eryiğit, Tuğba Pamay, et al. 2016a. IMST: A revisited Turkish dependency treebank. In *Proceedings of TurCLing 2016, the 1st International Conference on Turkic Computational Linguistics*. EGE UNIVERSITY PRESS.
- Umut Sulubacak, Memduh Gokirmak, Francis Tyers, Çağrı Çöltekin, Joakim Nivre, and Gülşen Eryiğit. 2016b. Universal Dependencies for Turkish. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3444–3454, Osaka, Japan.
- Almeida Jacqueline Toribio and Barbara E Bullock. 2012. *The Cambridge handbook of linguistic code-switching*. Cambridge University Press.
- Lin Wang and Haitao Liu. 2013. Syntactic variations in Chinese–English code-switching. *Lingua*, 123:58–73.
- Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droганova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 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, August. Association for Computational Linguistics.
- Yuan Zhang, Jason Riesa, Daniel Gillick, Anton Bakalov, Jason Baldridge, and David Weiss. 2018. A fast, compact, accurate model for language identification of codemixed text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 328–337, Brussels, Belgium, October-November. Association for Computational Linguistics.
- Lilja Øvrelid and Petter Hohle. 2016. Universal Dependencies for Norwegian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, may.

Dependency Parser for Bengali-English Code-Mixed Data enhanced with a Synthetic Treebank

Urmi Ghosh

MT & NLP Lab, KCIS

LTRC, IIIT-H

Hyderabad, India

urmi.ghosh@research.iiit.ac.in

Dipti Misra Sharma

MT & NLP Lab, KCIS

LTRC, IIIT-H

Hyderabad, India

dipti@iiit.ac.in

Simran Khanuja

BITS, Pilani

KK Birla Goa Campus

Goa, India

khanuja.simran7@gmail.com

Abstract

The development of code-mixing (CM) NLP systems has significantly gained importance in recent times due to an upsurge in the usage of CM data by multilingual speakers. However, this proves to be a challenging task due to the complexities created by the presence of multiple languages together. The complexities get further compounded by the inconsistencies present in the raw data on social media and other platforms. In this paper, we present a neural stack based dependency parser for CM data of Bengali and English by utilizing pre-existing resources for closely related Hindi and English CM treebank as well as monolingual treebanks for Bengali, Hindi and English. To address the issue of scarcity of annotated resources for Bengali-English CM pair, we present a rule based system to computationally generate a synthetic code-mixing treebank for Bengali and English (Syn-BE) which is used to further improve the accuracy of our dependency parser. For evaluation purpose, we present a dataset of 500 Bengali-English tweets annotated under Universal Dependencies scheme.

1 Introduction

Code-mixing refers to the mixing of various linguistic units (morphemes, words, modifiers, phrases, clauses and sentences) primarily from two participating grammatical systems within a sentence (Bhatia and Ritchie, 2008). This is essentially different from code-switching which refers to the co-occurrence of speech extracts belonging to two different grammatical systems (Gumperz, 1982). The occurrence can be both inter-sentential or intra-sentential, however there are strict phrasal boundaries and within one lexical unit, the syntax of only one language is maintained. Since the more recent works have not focused on the differences between the two phenomena, we will use these two terms interchangeably.

Recently, code-mixing which was often only observed in speech, has pervaded almost all forms of communication due to the growing popularity and usage of social media platforms by multilingual speakers (Rijhwani et al., 2017). Therefore, there has been considerable effort in building CM NLP systems such as language identification (Nguyen and Dogruoz, 2013; Solorio et al., 2014; Barman et al., 2014; Rijhwani et al., 2017), normalization and back-transliteration (Dutta et al., 2015). Part-of-speech (POS) and chunk tagging for code-mixing data for various South Asian languages with English have been attempted with promising results (Sharma et al., 2016; Nelakuditi et al., 2016). Ammar et al. (2016) developed a single multilingual parser trained on multilingual set of treebanks that outperformed monolingually-trained parsers for several target languages. In the CoNLL 2018 shared task, several participating teams developed multilingual dependency parsers that integrated cross-lingual learning for resource-poor languages and were evaluated on monolingual treebanks belonging to 82 unique languages (Zeman et al., 2018). However, none of these multilingual parsers have been evaluated on code-mixed data or adapted specifically for CM parsing.

The Bengali-English code-mixing is found in abundance as Bengali is widely spoken in India and Bangladesh. It is the second most widely spoken language in India after Hindi (Bhatia, 1982). Because of inherent structural and semantic similarity between Bengali and Hindi, we observe a close proximity between Bengali-English and Hindi-English code-mixing as well. Both of these language pairs deal with

the challenges of mixing different typologically diverse languages; SOV word order¹ for Hindi/Bengali and SVO word order for English. A dependency parser for Hindi-English code-mixing has been presented by Bhat et al. (2018). In comparison, Bengali-English code-mixing is left relatively unexplored barring significant works on language identification (Das and Gambäck, 2014) and POS tagging (Jamatia et al., 2015) which serve as preliminary tasks for more advanced parsing applications down the pipeline. The main hindrance to the development of parsing technologies for Bengali-English stem from the lack of annotated resources for the code-mixing of this language pair. In this paper, we try to utilize the pre-existing resources for widely available monolingual Bengali, Hindi and English as well as Hindi-English code-mixing and adapt them for Bengali-English dependency parsing. We also propose a rule based system to synthetically generate Bengali-English code-mixing data. An attempt has been made to generate code-mixing data for the Spanish-English language pair (Pratapa et al., 2018) but none for the Hindi-English or Bengali-English language pair as these pairs pose special challenges due to their different word orders which commonly violate most code-mixing theories (Sinha and Thakur, 2005). We further present a method to project dependency annotations to our Bengali-English CM data from monolingual Bengali and Hindi-English CM treebank and generate a synthetic treebank for Bengali-English (Syn-BE) which helps improve the accuracy of our dependency parser. For evaluation purpose, we present a dataset of 500 Bengali-English tweets annotated under Universal Dependencies scheme.

2 Universal Dependencies for Bengali-English

2.1 Data Preparation and Annotation

We prepared a dataset of 500 Bengali-English tweets by crawling over Twitter using Tweepy² - an API wrapper for Twitter. We identify the Bengali-English tweets by running the tweets through a language identification system (Bhat et al., 2018) trained on the dataset provided by ICON 2015.³ We select only those tweets which satisfy a minimum code-mixing ratio of 30:70(%). Here, code-mixing ratio is defined as:

$$\frac{1}{n} \sum_{s=1}^n \frac{E_s}{M_s + E_s}$$

where n is the number of sentences in the dataset, M_s and E_s are the number of words in the matrix and embedded language in sentence s respectively. Next, we manually select 500 tweets from the resulting tweets and normalize and/or transliterate each word before annotating them using Universal Dependency guidelines (Nivre et al., 2016) for POS and dependency tags. The language tags are annotated based on the tag set defined in (Solorio et al., 2014; Jamatia et al., 2015).

Figure 1 illustrates the conventions followed by our annotators for unique code-mixed constructions. Bengali verbification of English verb *start* by adding a Bengali light verb *hobe* (“will be”) leads to a *hybrid* compound verb *start hobe* (“will be”). Here, *start* is POS tagged as ‘NOUN’ instead of ‘VERB’ as it functions as a noun in this CM lexical unit and verbal inflection is observed only by the light verb *hobe* (“will be”). Also, #BOSS2 is tagged as ‘PROPN’ instead of ‘X’ as it is a syntactic token in this context. These annotations are consistent with the annotations for Hindi-English CM (Bhat et al., 2018).

The resulting dataset is split into three sets consisting of 200 tweets for testing, 160 for tuning and a third set of 140 tweets to be used as the training set in our stacking model for dependency parsing. The Bengali-English CM dataset is available at https://github.com/urmig/UD_bn-en.

2.2 Code Mixing Data Synthesis

Based on the token-level data distribution in Table 1, we observe that the matrix language in the majority of CM sentences is Bengali. The same is observed for the Hindi-English CM Data (Sharma et al., 2016). With this assumption, we proceed with the synthetic data generation by mixing English linguistic elements into the matrix of Bengali sentences. A frequently observed phenomenon in CM data is replacement of noun phrases in one language by the corresponding noun phrase in the other language

¹Subject, Object and Verb Order in transitive sentences

²<http://www.tweepy.org/>

³<http://ltrc.iit.ac.in/icon2015/>

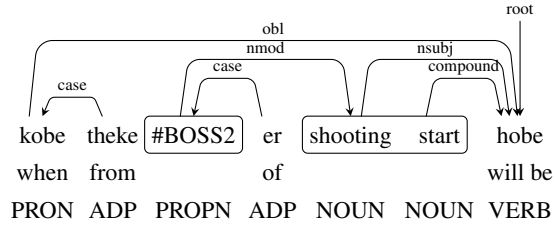


Figure 1: An example to illustrate Bengali-English Code-Mixed tweets

Language Tags	Token Count
Bengali	2840 (46.73%)
English	1781 (29.3%)
Rest (univ,acro,ne)	1457 (23.97%)

Table 1: Token-level Data Distribution on 500 Bengali-English tweets.

Language	POS	UAS	LAS
Hindi	97.65	94.36	91.02
Bengali	93.26	87.07	80.1
Hindi-English	91.90	74.16	64.11

Table 2: POS and parsing results of neural-stacking model for different languages

partially or in entirety (Dey and Fung, 2014). Sinha and Thakur (2005) had previously discussed CM constraints for Hindi-English and came to the conclusion that the phenomenon of code-mixing for this language pair is not entirely arbitrary. In our code-mixing method, we will be closely following *the Closed Class Constraint* which states that the matrix language elements within the closed class of grammar (possessives, ordinals, determiners, pronouns) are not allowed in code-mixing (Sridhar and Sridhar, 1980; Joshi, 1982).

- (1) **Bengali:** (*Apnar* “your” PRON) (*chokher* “of eyes” NOUN) (*dehashonar* “care” VERB) (*jonye* “for” ADP) (*aapni* “you” PRON) (*kotota* “how much” DET) (*icchuk* “willing” ADJ) ?
- (2) **English:** (*How* ADV) (*aware* ADJ) (*are* VERB) (*you* PRON) (*about* ADP) (*the* DET) (*care* NOUN) (*of* ADP) (*your* PRON) (*eyes* NOUN) ?
- (3) **Incorrect CS:** (**Your* PRON) (*chokher* “of eyes” NOUN) (**about* ADP) (**the* DET) (*dehashona* “care” NOUN) (**you* PRON) (**how* ADV) (*icchuk* “willing” ADJ)?
- (4) **Correct CS:** (*Apnar* “your” PRON) (*eyes* NOUN) (*er* “of” ADP) (*care* NOUN) (*er* “of” ADP) (*jonno* “for” ADP) (*aapni* “you” PRON) (*kotota* “how much” DET) (*aware* ADJ) ?

Example (3) demonstrates an unnatural and uncommon code-mixed construction and thus we can conclude that the two mixing constraints hold true for Bengali-English CM text as well. We extend these constraints to *question words* which can fall in the POS category of ADV and PRON as well as for *adpositions* (prepositions and postpositions). We note that the example (4) results in an acceptable code-mixed sentence as the closed class elements from the matrix language Bengali are retained.

The Code-Mixing Process

The pipeline for our code-mixing script is as shown in Figure 2. The script takes shallow-parsed English and Bengali parallel corpora as inputs. Consistency across chunks in parallel sentences is imperative for direct replacement of chunks for code-mixing. However, there are various structural differences in constituency parsing obtained for English by the Stanford Parser (Klein and Manning, 2003) and shallow parsing obtained for Bengali by the Shallow Parser by TDIL Program, Department Of IT Govt. Of India.⁴ The first module, *chunk harmonizer* handles the issue of structural differences in English and

⁴<http://ltrc.iiit.ac.in/analyzer/bengali/>

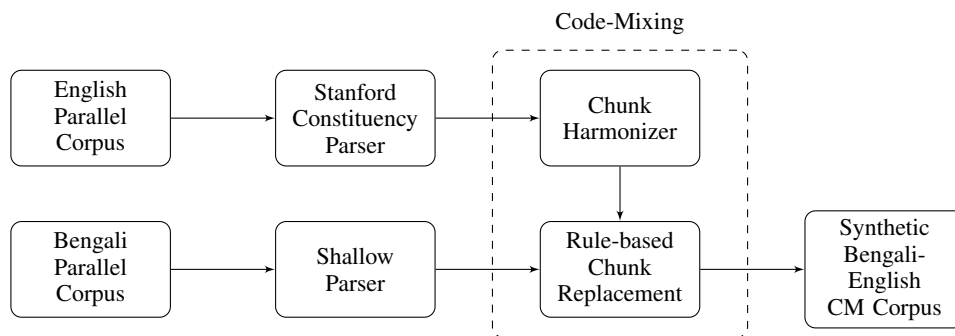


Figure 2: Schematic diagram of the Code-Mixing process

Bengali chunks by modifying the English chunks based on the following set of rules:

1. Separate the *coordinating conjunction* and its conjuncts into different chunks as they are treated separately in Bengali.
2. Combine the *adverbs of degree* (also, too, so, very etc.) with the preceding noun phrase (NP) as they are classified in Bengali as *particles* (*o* (“too”), *i* (“only”) etc.) and *intensifiers* (*bhishon* (“extreme”), *khub* (“very”) etc.) and grouped with NP.
3. Convert *prepositional phrase* (PP) to NP by making the head noun of the succeeding NP as the head and separating it from the preceding verb phrase (VP).
4. Split NP at *genitives* into separate NPs as genitives are considered as separate chunks in Bengali.

The rules are demonstrated by the example below:

(5) (NP *Your self-confidence*) (ADVP *also*) (VP *increases* (PP *with* (NP *teeth*))) → (NP *Your*) (NP *self-confidence also*) (VP *increases*) (NP *with teeth*)

which now consistently maps to the corresponding chunks in the parallel Bengali sentence:

(6) (NP *daanter* “teeth” *jonyo* “for”) (NP *aapnaar* “your”) (NP *aatmaviswas* “self-confidence” *o* “also”) (VP *baadhe* “increases”)

Along with harmonizing the chunks, this module marks the heads of each chunk in both languages using generalized rules defined by Sharma et al. (2006). For clarity, we have mapped the POS tags from Penn Treebank POS tagsets (Marcus et al., 1993) for English and Bureau Of Indian Standard (BIS) POS tagset (Choudhary and Jha, 2011) for Bengali to the Universal Dependency Tagset (Nivre et al., 2016).

The second module in the pipeline facilitates *rule-based chunk replacement* by taking the chunk-harmonized parallel Bengali and English sentences as inputs and replacing some selected Bengali chunks with English according to the rules discussed in 2.2. First, the chunks, each represented by the head element, are aligned using word alignments obtained from Giza++ (Och and Ney, 2003). Next, we replace the Bengali noun chunks (NP) and adjectival chunks (JJP) with the corresponding English chunks. By keeping the verbal chunks (VP) intact, we ensure that Bengali is retained as the matrix language of the code-mixed sentence. Hybrid compound verbs (see section 2.1) are a common occurrence in Bengali-English code-mixing and we can successfully synthesize them by replacing the NP/JJP preceding Bengali light verbs. For eg: (JJP *porishkaara* (“clean”)) (VP *koruna* (“do”)) → (JJP *clean*) (VP *koruna* (“do”)). We also retain Bengali post-positions and drop English prepositions associated with the heads.

Mixing the Bengali sentence (6) with the parallel English sentence (5) will generate:

(7) (NP *teeth er* “of” *jonyo* “for”) (NP *aapnaar* “your”) (NP *self-confidence o* “also”) (VP *baadhe* “increases”)

This is one of the acceptable combinations of the two sentences to form a CM sentence. We use the parallel corpora for English, Bengali and Hindi provided by Indian Languages Corpora Initiative (ILCI) (Jha, 2010) belonging to the *health* domain. We select a subset of 10,000 parallel sentences from each language and generate code-mixed sentences for both Bengali-English and Hindi-English language pair following the constraints in 2.2 . Thus, we have a parallel corpora for code-mixed Bengali-English and Hindi-English along with parallel corpora for Bengali, Hindi and English. We obtain only 5,063 code-mixed sentences with a minimum CM ratio of 30:70(%). The reason for this is attributed to the non-alignment of a few heads in many Bengali and Hindi sentences to the heads of corresponding English sentence. In spite of strictly following these rules, we generated a few erroneous sentences with word repetitions due to inconsistent chunking of multi-word expressions. We try to mitigate those errors in the post-processing step by carefully removing repeated words at code-mixing points. We attain this by calculating cosine similarity between the words represented by their *cross lingual embeddings* (see section 4). Eg: *chiniyukta* (“sugared”) sugared gums → sugared gum

2.3 Synthetic Bengali-English Treebank

Cross-lingual annotation projection makes use of parallel data to project annotations from the source language to the target language through automatic word alignment. Hwa et al. (2002) proposed some basic projection heuristics to deal with different kinds of word alignments. Tiedemann (2014) proposed improvements in the annotation scheme by adding heuristics to remove unnecessary dummy nodes that are introduced in the target treebank to deal with problematic word alignments. We investigate the utility of annotation projection from the Hindi-English CM treebank (HE) and the Bengali monolingual treebank (B) to Bengali-English (BE). HE is created by parsing the Hindi-English CM data generated in the section 2.2 using the neural stacking dependency parser for Hindi-English by Bhat et al. (2018).⁵ BE is generated by parsing the parallel Bengali sentences using the same neural stacking dependency parser trained on a monolingual Bengali dependency treebank. The POS tagging and parsing accuracy of these two parsers are mentioned in Table 2.

The basic setup for annotation projection is as follows:

1. Project annotations from B to BE for the matching head word nodes in Bengali and its dependent Bengali nodes.
2. Project annotations from HE to BE for the matching head word nodes in English and its dependent English nodes.
3. For each matching English dependent node in HE and BE with a Hindi head, find the aligned Bengali node in B. If the cosine similarity between the two is above a certain threshold (0.5), project annotations from B to BE.
4. For each matching Bengali dependent node in B and BE with an English head, find the aligned Hindi node in HE. If the cosine similarity between the two is above a certain threshold (0.5), project annotations from HE to BE.

In Figure 3, we demonstrate this with an example where the annotation for the BE tree is generated by both HE (in blue) and B (in red). Since the sentences in BE, HE and B are essentially parallel, we get one-to-one mapping and do not need to introduce any dummy nodes. We select 3643 completely annotated trees for our Syn-BE.

3 Dependency Parsing

We adapt the neural dependency parser by Bhat et al. (2018) which is based on a transition-based parser (Kiperwasser and Goldberg, 2016) and enhanced by neural stacks to incorporate monolingual syntactic knowledge with the CM model. The model jointly learns POS-tagging as well as parsing by adapting feature level neural stacks (Zhang and Weiss, 2016; Chen et al., 2016). The input layer for both the

⁵<https://github.com/irshadbhat/csnlp>

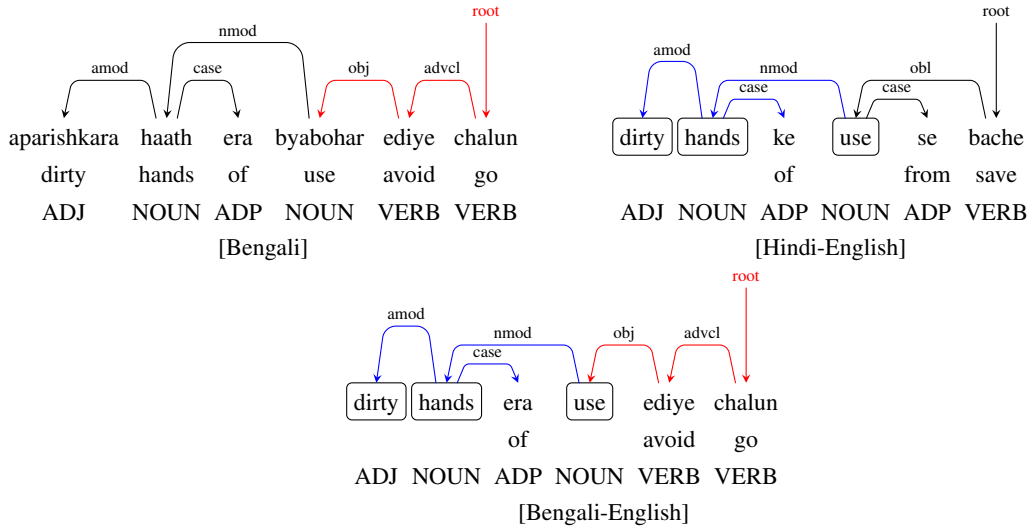


Figure 3: An example of annotation projection from Bengali and Hindi-English to Bengali-English

tagger and the parser encodes the input sentence into word and character embeddings and passes it to the shared bidirectional LSTM (Bi-LSTM). Bhat et al. (2018) demonstrates augmenting the final multi-layer perceptron (MLP) layer of a bilingual model trained on Hindi and English treebanks (bilingual source model) into the MLP layer of the model trained on Hindi-English CM data (CM model) achieves state-of-the-art results for Hindi English code-mixing.

4 Experiments

Our models are trained on English and Hindi UD-v2 treebanks.⁶ Due to the absence of a Bengali UD treebank, we converted the Paninian annotation scheme (Begum et al., 2008) present in the Bengali treebank⁷ to UD by slightly modifying the rules (Tandon et al., 2016) for Hindi. The characters are represented by 32-dimensional character embeddings while the words in each language are represented by 64 dimensional word2vec vectors (Mikolov et al., 2013) learned using the skip-gram model. The hidden dimensions and learning hyperparameters are consistent with those in Bhat et al. (2018).

For our baseline model, we train the neural stacking model (Bhat et al., 2018) for Bengali-English by training the source model on both Bengali and English treebanks and stacking it on a CM model trained on 140 Bengali-English CM (Gold-BE) sentences in our training set. Even though the size of the training set is limited, we benefit from the presence of unique CM grammar as well as syntactic information of social media elements. Our bilingual source model serves to transfer both POS tagging and parsing information to the CM model.

In our next experiment, we train the CM stacking model with 1448 Hindi-English CM data (Gold-HE) as provided by Bhat et al. (2018) in addition to our 140 Gold-BE sentences. In order to fully capture the Hindi syntactic information in the CM data, we fortify the bilingual source model with the Hindi treebank resulting in a trilingual source model. We try to reduce the differences in data representations belonging to Hindi and Bengali by using:

1. *Cross Lingual Word Embeddings* for Hindi and Bengali by projecting the word2vec embeddings for the two languages into the same space by using the projection algorithm of Artetxe et al. (2016) and using a bilingual lexicon from ILCI parallel corpora.
2. *WX notation*⁸ to represent words from the two languages and using a common 32-dimensional character embedding space.

⁶<https://github.com/UniversalDependencies>

⁷Developed as a part of the Indian Languages Treebanking Project by Jadavpur University

⁸http://wiki.apertium.org/wiki/WX_notation

Embeddings	POS	UAS	LAS
Monolingual	84.86	71.32	56.93
Crosslingual	85.62	71.94	57.41
Crosslingual + WX notation	87.43	74.42	60.04

Table 3: Effect of embeddings on POS and Parser results for the Trilingual + Gold-(HE + BE) model

Stacking Models	POS	UAS	LAS
(Bilingual) + Gold-BE	79.39	62.78	49.38
(Trilingual) + Gold-(HE + BE)	87.43	74.42	60.04
(Trilingual + Syn-BE) + Gold-(HE + BE)	89.63	76.24	61.41

Table 4: POS and Parser results of different neural-stacking models for Bengali-English.

For our final experiment, we augment our Synthetic Code-Mixed Bengali-English Treebank (Syn-BE) to the trilingual source model generated in the previous experiment and stack that on our CM model.

5 Results

We present our final results in Table 4. The baseline model adapted from Bhat et al. (2018) for Hindi-English gives us 62.78% UAS and 49.38% LAS points. The POS results give 79.39% accuracy. The lower accuracy for the model is expected due to the small training set for Bengali-English (140) when compared with Hindi-English (1448). Moreover, the significantly lower parser accuracy (a difference of ~9% LAS points) for Bengali in comparison to Hindi negatively impacts the performance of the source model (See Table 2).

Our next model that fortifies the baseline model with Hindi monolingual and CM data with Hindi-English improves all the three measurements significantly because it enables us to utilize the relatively large Hindi-English CM UD-annotated data. The UAS and LAS show an improvement in accuracy by 11.64% and 10.66% points respectively. The improvement in POS accuracy is ~8%. In this model, we slightly modify the word and character embedding representations in order to mitigate the lexical differences between Hindi and Bengali by using cross-lingual embeddings and a common character space. From Table 3, we observe that using cross-lingual embeddings improves the accuracy of tagging by 0.76%, UAS by ~0.6% points and LAS by ~0.5% points. Using a common character space by using WX notation further improves the accuracy of both tagging and parsing by ~1.8% and ~2.5% points respectively. The significant improvements in the results confirm the inherent similarity between the code-mixing grammar of Hindi and Bengali with English as both of these language pairs deal with mixing of two typologically diverse languages.

Our final model utilizes our Syn-BE CM treebank by augmenting it to the trilingual source model and stacking it on the CM model trained on our Gold-HE and Gold-BE datasets. We observe an improvement in the Bengali-English parser accuracy by 1.82% UAS points, 1.37% LAS points and POS tagging accuracy by 2.2%. This improvement is satisfactory considering the errors propagated into our Syn-BE treebank by annotating projections from automatically parsed Bengali and Hindi-English treebanks. We must also note that the the domain of Syn-BE (*health*) lacks certain social media elements and constructs present in the evaluation set.

6 Conclusion

Our neural stacking model utilizing monolingual, gold and synthetic CM resources has shown significant improvement of 10.24% for POS, 13.76% improvement in UAS and ~12% improvement in LAS points when compared with the baseline model. The stacking model augmented by the Syn-BE CM treebank improves the POS tagging accuracy by 2.2% points and parser accuracy by 1.82% UAS points and 1.37% LAS points respectively. The Syn-BE CM data can be used in other NLP systems like machine translation, question-answering etc. to further improve their systems. There is scope for extending the Syn-BE corpus by including more CM constructions like intra-sentential switching and CM sentences with English as the matrix language. Our evaluation dataset consisting of 500 UD-annotated Bengali-English tweets provides for a valuable resource for research on code-mixing.

References

- Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.
- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294.
- Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the first workshop on computational approaches to code switching*, pages 13–23.
- Rafiya Begum, Samar Husain, Arun Dhvaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency annotation scheme for indian languages. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.
- Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Manish Shrivastava, and Dipti Misra Sharma. 2018. Universal dependency parsing for hindi-english code-switching. *arXiv preprint arXiv:1804.05868*.
- Tej Bhatia and William Ritchie. 2008. *The Handbook of Bilingualism*. 01.
- TEJ K. Bhatia. 1982. English and the Vernaculars of India: Contact and Change1. *Applied Linguistics*, III(3):235–245, 10.
- Hongshen Chen, Yue Zhang, and Qun Liu. 2016. Neural network for heterogeneous annotations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 731–741.
- Narayan Choudhary and Girish Nath Jha. 2011. Creating multilingual parallel corpora in indian languages. In *Language and Technology Conference*, pages 527–537. Springer.
- Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387.
- Anik Dey and Pascale Fung. 2014. A hindi-english code-switching corpus. In *LREC*, pages 2410–2413.
- Sukanya Dutta, Tista Saha, Somnath Banerjee, and Sudip Kumar Naskar. 2015. Text normalization in code-mixed social media text. In *2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS)*, pages 378–382. IEEE.
- John J. Gumperz. 1982. *Discourse Strategies*. Studies in Interactional Sociolinguistics. Cambridge University Press.
- Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak. 2002. Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 392–399. Association for Computational Linguistics.
- Anupam Jamatia, Björn Gambäck, and Amitava Das. 2015. Part-of-speech tagging for code-mixed english-hindi twitter and facebook chat messages. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 239–248.
- Girish Nath Jha. 2010. The tdil program and the indian language corpora initiative (ilci). In *LREC*.
- Aravind K Joshi. 1982. Processing of sentences with intra-sentential code-switching. In *Proceedings of the 9th conference on Computational linguistics-Volume 1*, pages 145–150. Academia Praha.
- Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.
- Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.
- Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank.

- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Kovida Nelakuditi, Divya Sai Jitta, and Radhika Mamidi. 2016. Part-of-speech tagging for code mixed english-telugu social media data. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 332–342. Springer.
- Dong-Phuong Nguyen and A. Seza Dogruoz. 2013. Word level language identification in online multilingual communication. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 857–862, United States, 10. Association for Computational Linguistics (ACL).
- 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 Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553.
- Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Shekhar Maddila. 2017. Estimating code-switching on twitter with a novel generalized word-level language detection technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1971–1982, Vancouver, Canada, July. Association for Computational Linguistics.
- Dipti Misra Sharma, Rajeev Sangal, Lakshmi Bai, Rafiya Begam, and KV Ramakrishnamacharyulu. 2006. Anncorra: Treebanks for indian languages (version-1.9). Technical report, Technical report, Language Technologies Research Center IIIT, Hyderabad, India.
- Arnav Sharma, Sakshi Gupta, Raveesh Motlani, Piyush Bansal, Manish Srivastava, Radhika Mamidi, and Dipti M Sharma. 2016. Shallow parsing pipeline for hindi-english code-mixed social media text. *arXiv preprint arXiv:1604.03136*.
- R Mahesh K Sinha and Anil Thakur. 2005. Machine translation of bi-lingual hindi-english (hinglish) text. *10th Machine Translation summit (MT Summit X)*, Phuket, Thailand, pages 149–156.
- Tamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72.
- Shikaripur N Sridhar and Kamal K Sridhar. 1980. The syntax and psycholinguistics of bilingual code mixing. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 34(4):407.
- Juhi Tandon, Himani Chaudhry, Riyaz Ahmad Bhat, and Dipti Sharma. 2016. Conversion from paninian karakas to universal dependencies for hindi dependency treebank. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 141–150.
- 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.
- 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, October. Association for Computational Linguistics.
- Yuan Zhang and David Weiss. 2016. Stack-propagation: Improved representation learning for syntax. *arXiv preprint arXiv:1603.06598*.

tweeDe – A Universal Dependencies treebank for German tweets

Ines Rehbein
Leibniz ScienceCampus
Heidelberg University/
IDS Mannheim
{rehbein|ruppenhofer}@ids-mannheim.de

Josef Ruppenhofer
Institut für
Deutsche Sprache
Mannheim

Bich-Ngoc Do
Leibniz ScienceCampus
Heidelberg University/
IDS Mannheim
do@cl.uni-heidelberg.de

Abstract

We introduce the first German treebank for Twitter microtext, annotated within the framework of Universal Dependencies. The new treebank includes over 12,000 tokens from over 500 tweets, independently annotated by two human coders. In the paper, we describe the data selection and annotation process and present baseline parsing results for the new test suite.

1 Introduction

Recent years have seen an increasing interest in developing robust NLP applications for data from different language varieties and domains. The Universal Dependencies (UD) project (Nivre et al., 2016) has inspired the creation of many new datasets for dependency parsing in a multilingual setting. Treebanks have been created for low-resourced languages such as Bambara, Erzya, or Kurmanji as well as for many new domains, genres and language varieties for which no annotated data was yet available. A case in point are web genres, spoken discourse, literary prose, historical data or data from social media.¹

We contribute to the creation of new resources for different language varieties and introduce tweeDe, a new German UD Twitter treebank. TweeDe has a size of over 12,000 tokens, annotated with PoS, morphological features and syntactic dependencies. TweeDe is different from existing German UD treebanks as its content focusses on private communication. Private tweets share many properties of spoken language. They are often highly informal and not carefully edited, often lack punctuation and can include ungrammatical structures. In addition, the data often includes spelling errors and a creative use of language that results in a high number of unknown words. These properties make user-generated microtext a challenging test case for parser evaluation.

In the paper, we describe the creation of tweeDe, including data selection, preprocessing and the annotation process. We report inter-annotator agreement for the syntactic annotations (§2) and discuss some of the decisions that we have made during annotation (§3). We compare tweeDe to other treebanks in §4. In §5 we present baseline parsing results for the new treebank. Finally, we put our work into context (§6) and outline avenues for future work (§7).

2 tweeDe – A German Twitter treebank

This section describes the creation of the first German Twitter treebank, annotated with Universal Dependencies. The treebank includes 519 tweets with over 12,000 tokens of microtext.

2.1 Data extraction

The annotation of user-generated microtext is a challenging task, due to the brevity of the messages and the missing context information, which often results in highly ambiguous texts. As a result, inter-annotator agreement (IAA) is often below the one obtained on standard newspaper text. To avoid such problems, we opted to extract short communication threads, which range in length from 2 up to 34 tweets. This approach allowed the annotators to see the context of each tweet and was thus crucial for resolving ambiguities in the data.

¹The different treebanks and their description are available from: <https://universaldependencies.org/>.

The conversations were collected in two steps. We first used an existing python tool² that supports the downloading of conversations by querying the Twitter API for a set of query terms and then scraping the html page on twitter.com that represents each matching conversation. However, Twitter does not embed complete json files into the html-pages and the existing crawler had some problems in fully retrieving tweet text containing certain special characters. We therefore used the output of the initial crawler only to establish the ids and the sequencing of the tweets in a conversation and then re-downloaded the full json files to be sure we had complete tweets.

The query terms we used were all German stop words, i.e. highly-frequent closed-class function words such as prepositions, articles, modal verbs, and adverbs such as *auch* ‘too’ or *dann* ‘then’. The idea behind this was to avoid any kind of topic bias. Of the threads retrieved, we only retained those representing private communication between two or more participants. Threads consisting mainly of automatically generated tweets, advertisements, and so on were discarded after manual inspection. The treebank preserves the temporal order of the tweets in the same thread. For meta-information, we keep the tweet id, date and time as well as the author’s user name. As is common practise for UD treebanks, we also store the raw, untokenised text for each tweet.

Besides issues arising from brevity, further problems for annotating user-generated social media content are the creative use of language, including acronyms (example 1) and emoticons (example 2), non-canonical spellings (example 3), missing arguments (example 2) and the often missing or inconsistent use of punctuation (examples 1-4). The latter causes segmentation problems like those faced in annotating spoken language where, since no punctuation is given, the annotator has to decide on where to insert sentence boundaries.

- | | |
|--|--|
| <p>(1) hdl
have you dear
“Love you”</p> | <p>(2) Mache deshalb gerne mal mit < 3
participate thus gladly MODAL PTCL VERB PTCL EMOTICON
“Hence (I) like to participate once in a while < 3”</p> |
| <p>(3) Is nich wahr ich habe nur einen report bekommen das sie es erhalten haben und überprüfen..
is not true I have only a report got that they it received have and check..
“It’s not true. I only got a report that they have received it and will check it.”</p> | |
| <p>(4) Mahlzeit Arbeit Gassigang Wohnung geputzt Essen gemacht Jaaaa es ist #Freitag und jetzt
meal work walking the dog flat cleaned food made Yeeees it is Friday and now
#hochdiehaendewochenende
#up-the-hands-weekend</p> | |

2.2 Segmentation

For spoken German, several proposals have been made how to segment transcribed utterances, based on syntax, intonation and prosodic cues, pausing and hesitation markers (Rehbein et al., 2004; Selting et al., 2009). However, when the different levels of analysis provide contradicting evidence, it is not clear how to proceed. For tweets, we have to deal with similar issues. When no (or only inconsistent use of) punctuation is present, we have to decide how to segment the tweet into units for syntactic analysis. Earlier work has chosen to consider the whole tweet as one unit, i.e. as one syntax tree. Since Twitter has changed their policy and doubled the length limit from 140 to 280 characters, this is no longer feasible (see example 5 below). We thus decided to split up the messages into sentences, based on the following rules.

- (5) @surfguard @Mathias59351078 @ArioMirzaie Über einige amüsiere ich mich köstlich, bei manchen denke ich "hm" und bei wieder anderen bin ich entsetzt. Mit keinem einzigen hab ich irgendwas zu tun. Wenn du mich wegen meiner Hautfarbe den Schuldigen zuordnest, bist du ein Rassist.

“@surfguard @Mathias59351078 @ArioMirzaie Some make me laugh, some make me think "hm" and still others make me feel appalled. I don’t have anything to do with any of them. If you blame me for the color of my skin, you’re a racist.”

- Hashtags and URLs at the beginning or end of the tweet that are not syntactically integrated in the sentence are separated and form their own unit (tree).
- Emoticons are treated as non-verbal comments to the text and are integrated in the tree (figure 1).

²<https://github.com/song9446/twitter-corpus-crawler-python>

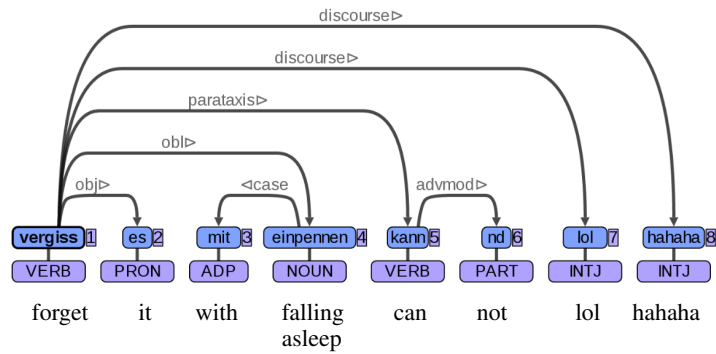


Figure 1: Example tree from tweede, displayed in UD-annotatrix (Tyers et al., 2018).

- Interjections (*Aaahh*), inflectives (**grins**), fillers (*ähm*) and acronyms typical for social media content (*lol*, *OMG*) are also not separated but considered to be part of the tree (figure 1).

2.3 Tokenisation

User-generated text often reflects (or mimics) morpho-phonological processes from spoken language that are in conflict with the rules of Standard German orthography. One example are words merged into one token that, according to German grammar, should be separated but in spoken varieties of German are contracted into one token. We split merged tokens to avoid having tokens with more than one PoS tag and grammatical function. To mark that the word has been written as one atomic token, we use the UD feature `SpaceAfter=No` in combination with `CorrectSpaceAfter=Yes` in the last column of the CoNLL-UD file. Figure 2 (left) shows an example where the canonical token sequence “Kennst Du ?” is instead fused into the single token “Kennste ?”.

We also observe the opposite case where tokens that should have been written as one word are split into two or more separate tokens in the tweet. Most of these are German noun compounds. We chose to annotate split compounds using the UD relation `goeswith`. We follow UD conventions to always annotate the first component as the head and attach all remaining components to the first component. One problem with this approach is that in some cases the head of the compound will end up with the wrong PoS tag. Figure 2 (right) gives an example where the whole compound should have been annotated as a noun (*Japanurlaub*, Japan vacation) but instead now obtains a proper noun PoS tag. A possible solution to this problem is to deviate from UD practise and annotate the second component (i.e. the real head) as the head. As those cases were rare in our data, we refrained from doing so, for the sake of consistency with other UD treebanks.

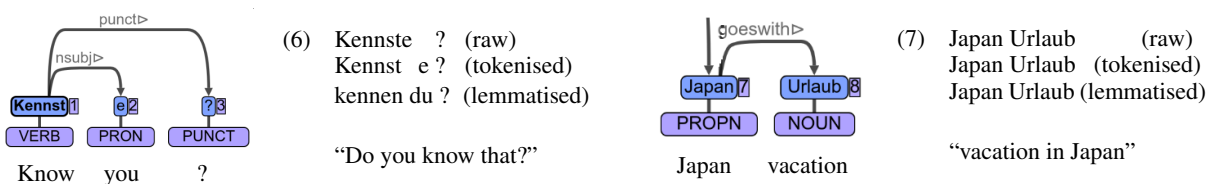


Figure 2: Merged tokens (left) and split compound (right)

2.4 Annotation

We annotated two types of PoS tags, based on the UD (Petrov et al., 2012) and Stuttgart-Tübingen (STTS) (Schiller et al., 1995) tag sets. The PoS tags and morphological features represent the annotations of one annotator, correcting the output of the UD processing pipeline for German (UDPipe) (Straka and Straková, 2017). For all dependency annotations, two annotators provided syntactic attachments and dependency labels, which were subsequently adjudicated. The adjudicated syntactic dependency relations were used for consistency checks between the dependency labels and the PoS and morphological tags. Additional consistency checks based on DECCA (Dickinson and Meurers, 2003) verified the compatibility of the different annotation layers. All incompatibilities were manually inspected and resolved. The final testsuite includes 12,073 tokens from 519 tweets, split up into train, development and test data (table 1). Around 10% of the tweets include a non-projective tree structure.

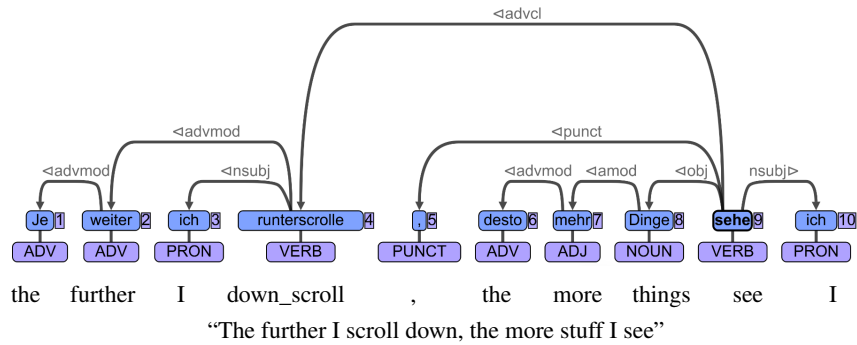


Figure 5: Comparative clause with *je-desto* in tweeDe.

Correlative construction with two clauses The correlative construction *je X, desto/umso Y* (the X, the Y) (figure 5) consists of a subordinate clause marked by *je*, followed by a matrix clause that is introduced by *desto/umso*.³ Each clause needs to contain a comparative form, either of an adjective or of an adverb. Semantically, the construction describes a relationship between an independent and a dependent variable (example 8).

As indicated by word order, the clause expressing the *causal variable* is the subordinate clause (the finite verb comes last) while the clause describing the *dependent variable* is syntactically encoded as the matrix clause (the finite verb comes in second position). While *je* typically only marks the subordinate clause, there also exist variants of the construction where the *desto/umso* is omitted and a second *je* is used instead to mark the comparative that describes the dependent variable (example 9).⁴

- (8) Je älter ich werde, umso glücklicher bin ich.
 PTCL older I become, PTCL happier am I.
 “The older I get, the happier I am.”
- (9) Je größer die Gruppe, je kleiner der Preis.
 PTCL bigger the group, PTCL smaller the price.
 “The larger the group, the smaller the price.”

Based on these observations, we decided to attach the subordinate clause as an adverbial clause to the matrix clause and analyse both particles as adverbial modifiers. We do not assign the mark relation as the particles are not modifiers of the head of the subordinate clause but are modifiers of the comparative forms in the subordinate and in the matrix clause.

This analysis is different from the one in the German UD-GSD and TüBa-D/Z UD treebanks (figure 6) where the head of the subordinate clause is analysed as the root of the sentence and the matrix clause is attached as a conjunct of the subordinate clause. Our analysis is consistent with the one for conditional clauses that are similar in meaning (e.g.: *If I scroll down further, I can see more*), where the subordinate if-clause is also an adverbial clausal modifier of the matrix clause.

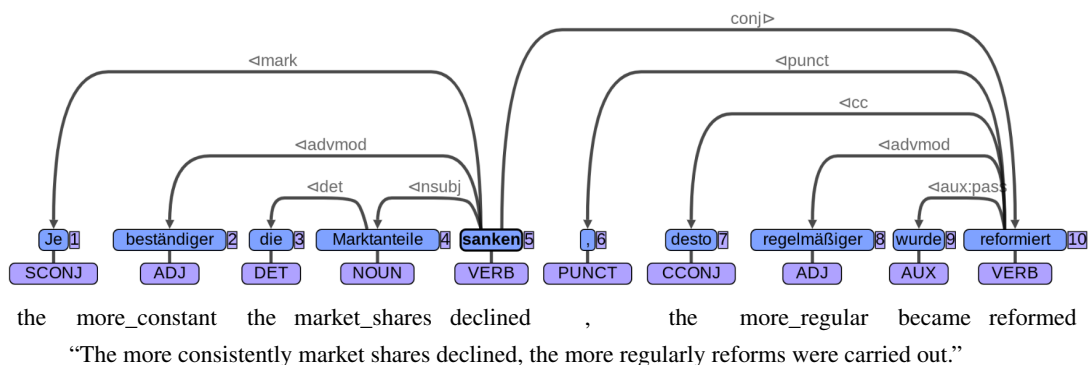


Figure 6: Comparative clause with *je-desto* in the TüBa-D/Z-UD.

4 Comparison to other German UD treebanks

We now compare tweeDe to three other German treebanks, i) UD-GSD, ii) TüBa-D/Z and iii) UD-HDT. The UD-HDT (Hennig and Köhn, 2017) is a conversion of the Hamburg Dependency Treebank (Foth et

³While this is the canonical order, it is also possible to switch the order of the matrix and subordinate clauses. Constructions without verbal predicates are also possible: *Je mehr, desto lustiger*. (The more, the merrier).

⁴While these are less frequent than the canonical form with *je-desto/umso*, it is easy to find instances in a large corpus such as the DeWac (Baroni et al., 2009), as well as instances that include only the *je* without a second particle where the matrix clause then needs to be in V1 word order.

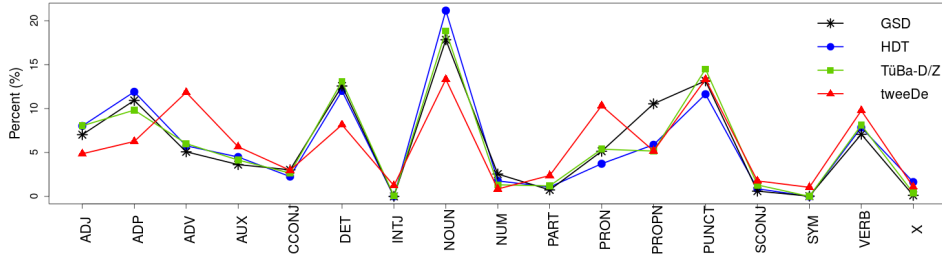


Figure 7: Distribution of UD PoS tags in four German UD treebanks.

al., 2014) which includes mostly news articles and is also the largest existing German treebank.

Figure 7 shows the distribution of PoS tags in the four treebanks. While the other three treebanks are quite homogeneous (except UD-GSD including more proper names), the most striking difference between tweeDe and the other treebanks is the higher number of adverbs and pronouns. This is typical for informal multiparty communication and is accompanied by a lower percentage of nouns, determiners, adjectives and adpositions as well as a slightly higher amount of verbs. This shows that tweeDe has a more verbal style, as opposed to the nominal style of the other treebanks.

5 Parsing experiments

We present parsing baselines for the new German UD treebank, using the state-of-the-art parser of Dozat et al. (2017). The parser is a neural dependency parser that learns complex, non-linear representations directly from the input text, based on bidirectional LSTMs (Hochreiter and Schmidhuber, 1997). It only considers local context and predicts attachments and labels in a greedy fashion. The huge success of the parser is based on its use of biaffine attention.

In our first experiment, we train the parser on the 250 tweets in the tweeDe training set. We use pretrained skipgram embeddings with 100 dimensions (window size: 5, min word count: 10), trained on a large collection of German tweets, collected in a time period from 2013 to 2017. The embeddings are publically available from <https://www.cl.uni-heidelberg.de/research/downloads>. All models have been trained with default parameters.

Table 2 (left) shows results for gold PoS and for automatically predicted PoS tags. Using UD PoS tags for parsing outperforms the STTS tags by a large margin, probably due to sparsity caused by the more fine-grained STTS. Feeding both, UD and STTS tags, to the parser can further increase results, but only slightly (less than 1%). Most surprisingly, we obtain higher results when using automatically predicted STTS tags (as compared to using gold STTS tags). This observation, however, is more pronounced for the test set and might not be representative, being an artefact of the small data size.

Results for training on the small tweeDe dataset only are in the range of 74% LAS (gold PoS) and 68% LAS (auto PoS). When adding the training data from the German-GSD UD treebank, results increase to 81% LAS (gold PoS) and 76% LAS (auto PoS). The large gap of 5% between the gold and auto PoS setting highlights the importance of high-quality PoS tags for parsing tweets.

	PoS tagset	dev		test	
		UAS	LAS	UAS	LAS
gold	UD	82.15	74.26	80.65	72.69
	STTS	73.48	63.05	70.28	60.83
	BOTH	82.51	74.94	81.51	74.34
auto	UD	78.88	69.90	76.01	67.08
	STTS	72.91	63.21	71.25	62.64
	BOTH	79.09	70.73	76.60	68.14

	PoS tagset	dev		test	
		UAS	LAS	UAS	LAS
gold	UD	88.17	81.73	86.40	80.47
	STTS	85.21	77.32	81.38	74.02
	BOTH	88.89	82.67	87.15	81.01
auto	UD	85.88	78.20	82.91	76.03
	STTS	84.90	76.44	82.32	74.79
	BOTH	86.30	78.15	83.31	76.39

Table 2: Parsing results for the Dozat parser on tweeDe, without (left) and with additional training data from the German-GSD UD treebank (right).

	# token	# tweets	LAS	(parser)
EN (Foster et al. 2011)	n.a.	519*	67.3	Malt2006
EN (Kong et al. 2014)	12,149	840	–	
EN (Liu et al. 2018)	55,607	3,550	77.7	D&M2017
EN-AAE (Blodgett et al. 2018)	3,072	250	56.1	D&M2017
EN-MS (Blodgett et al. 2018)	3,524	250	67.7	D&M2017
IT (Sanguinetti et al. 2018)	124,410	6,712	81.5	D&M2017

Table 3: Statistics for manually annotated treebanks (*Foster et al. only report # sentences, not # tweets. We expect the no. of tweets to be slightly lower than 500). The data of Blodgett et al. includes AAE and main-stream (MS) English tweets. The last two columns report results for the Dozat & Manning parser (Dozat et al., 2017) (w/o domain adaptation) or the Malt parser from the literature.

6 Related work

Twitter treebanks exist not only for English (Kong et al., 2014; Liu et al., 2018; Blodgett et al., 2018) but also for Italian (Sanguinetti et al., 2018) and Arabic (Albogamy et al., 2017). Foster et al. (2011) were among the first to provide syntactic analyses for Twitter microtext. They created a testset with over 500 sentences extracted from tweets. The data was automatically parsed with a constituency parser and the trees were manually corrected by one annotator. Inter-annotator agreement (IAA) for labelled bracketing, measured on a subset of the data annotated by a second annotator, was quite high with nearly 96%. Parsing accuracy without any domain adaptation, however, was low: the Malt parser (Nivre et al., 2006), trained on the WSJ, achieved an LAS of 63.3% on the Twitter testset.

The Tweepbank v1 (Kong et al., 2014) is another English Twitter treebank, with a size of over 900 tweets annotated with unlabelled dependencies. Liu et al. (2018) extend the work of Kong et al. (2014) by enlarging the treebank to more than 3,500 tweets, refining the guidelines and adding labels to the former unlabelled trees. They report an IAA of 84.3% for labelled attachments in the Tweepbank v2. A third English Twitter treebank was created by Blodgett et al. (2018). Their corpus includes 250 African-American English (AAE) tweets and 250 tweets of mainstream American English microtext. The data has been annotated by two coders but no inter-annotator agreement is reported.

The Italian Twitter treebank of Sanguinetti et al. (2018) is the largest existing Twitter treebank and includes more than 6,700 trees. The authors report an IAA of 0.92 κ for syntactic annotation. The results for a dependency parser (Dozat et al., 2017) trained on a combination of the Italian UD treebank and the new dataset are also quite high, with a labelled attachment score of 81.5%. The high agreement and parsing scores suggest that the dataset is somewhat easier and more well-behaved than the Tweepbank (see table 3 for baseline results for the different Twitter treebanks).

For Arabic, a treebank with Twitter microtext has been created fully automatically, based on predictions of a rule-based and a data-driven parser (Albogamy et al., 2017). Efforts have been made to map the annotations to the UD scheme, but, to the best of our knowledge, the data is not yet available.

With over 12,000 tokens, our new German Twitter treebank is comparable in size to TWEEBANK v1 (Kong et al., 2014) even though the number of tweets in our dataset is smaller. This is due to the fact that our data were collected after Twitter raised the maximum length for tweets from 140 to 280 characters.

7 Conclusions

We presented tweeDe, the first German Twitter treebank, as a new training and testsuite for UD parsing. tweeDe includes more than 12,000 tokens of informal private communication, annotated for PoS, morphology and UD syntactic dependencies. The data will be made available to the research community.⁵

We also presented parsing baselines for the new dataset, showing that combining a small amount of in-domain Twitter data in combination with a larger amount of out-of-domain data can yield parsing accuracies in the range of 83% (UAS) and 76% (LAS) on the new testsuite.

⁵<https://www.cl.uni-heidelberg.de/research/downloads>.

References

- Fahad Albogamy, Allan Ramsay, and Hanady Ahmed. 2017. Arabic tweets treebanking and parsing: A bootstrapping approach. In *The 3rd Arabic Natural Language Processing Workshop*, WANLP 2017, pages 94–99.
- Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.
- Su Lin Blodgett, Johnny Wei, and Brendan T. O’Connor. 2018. Twitter Universal Dependency Parsing for African-American and Mainstream American English. In *The 56th Annual Meeting of the Association for Computational Linguistics*, ACL 2018, pages 1415–1425, Melbourne, Australia.
- Çağrı Çöltekin, Ben Campbell, Erhard Hinrichs, and Heike Telljohann. 2017. Converting the TüBa-D/Z Treebank of German to Universal Dependencies. In *The NoDaLiDa 2017 Workshop on Universal Dependencies*, UDW 2017, pages 27–37.
- Markus Dickinson and W. Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, EACL 2003, pages 107–114.
- Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford’s Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30.
- Jennifer Foster, Özlem Çetinoglu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. #hardtoparse: POS tagging and parsing the twitterverse. In *Analyzing Microtext, Papers from the 2011 AAI Workshop, San Francisco, California, USA, August 8, 2011*.
- Kilian A. Foth, Arne Köhn, Niels Beuck, and Wolfgang Menzel. 2014. Because size does matter: The Hamburg Dependency Treebank. In *The 9th International Conference on Language Resources and Evaluation*, LREC 2014, pages 2326–2333.
- Felix Hennig and Arne Köhn. 2017. Dependency tree transformation with tree transducers. In *The NoDaLiDa Workshop on Universal Dependencies*, UDW@NoDaLiDa 2017, pages 58–66.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.
- Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archana Bhatia, Chris Dyer, and Noah A. Smith. 2014. A dependency parser for tweets. In *The 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2014, pages 1001–1012, Doha, Qatar.
- Yijia Liu, Yi Zhu, Wanxiang Che, Bing Qin, Nathan Schneider, and Noah A. Smith. 2018. Parsing tweets into universal dependencies. In *The 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT 2018, pages 965–975, New Orleans, Louisiana, USA.
- Ryan T. McDonald, Joakim Nivre, Yvonne Quirnbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B. Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *The 51st Annual Meeting of the Association for Computational Linguistics*, ACL 2013, pages 92–97.
- Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *The 5th International Conference on Language Resources and Evaluation*, LREC 2006, pages 2216–2219.
- 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 *The Tenth International Conference on Language Resources and Evaluation*, LREC 2016, pages 1659–1666.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *The 8th International Conference on Language Resources and Evaluation*, LREC 2012, pages 2089–2096, May.
- Jochen Rehbein, Thomas Schmidt, Bernd Meyer, Franziska Watzke, and Annette Herkenrath. 2004. *Handbuch für das computergestützte Transkribieren nach HIAT*. Sonderforschungsbereich 538.

- Manuela Sanguinetti, Cristina Bosco, Alberto Lavello, Alessandro Mazzei, Oronzo Antonelli, and Fabio Tamburini. 2018. PoSTWITA-UD: an Italian Twitter Treebank in Universal Dependencies. In *The 11th International Conference on Language Resources and Evaluation*, LREC 2018, pages 1768–1775.
- Anne Schiller, Simone Teufel, and Christine Thielen. 1995. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, Universität Stuttgart, Universität Tübingen.
- Margret Selting, Peter Auer, Dagmar Barth-Weingarten, Jörg R. Bergmann, Pia Bergmann, Karin Birkner, Elizabeth Couper-Kuhlen, Arnulf Deppermann, Peter Gilles, Susanne Günthner, et al. 2009. Gesprächsanalytisches Transkriptionssystem 2 (GAT 2). *Gesprächsforschung: Online-Zeitschrift zur verbalen Interaktion*.
- 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*, pages 88–99, Vancouver, Canada, August. Association for Computational Linguistics.
- Heike Telljohann, Erhard W. Hinrichs, and Sandra Kübler. 2004. The Tüba-D/Z Treebank: Annotating German with a Context-Free Backbone. In *The Fourth International Conference on Language Resources and Evaluation*, LREC 2004.
- Francis M. Tyers, Mariya Sheyanova, and Jonathan North Washington. 2018. Ud annotatrix: An annotation tool for universal dependencies. In *The 16th International Workshop on Treebanks and Linguistic Theories*, TLT 2016, pages 10–17.

Creating, enriching and valorising treebanks of Ancient Greek: the ongoing Pedalion-project

Alek Keersmaekers Research Foundation Flanders & KU Leuven alek.keersmaekers @kuleuven.be	Wouter Mercelis KU Leuven wouter.mercelis @student.kuleu- ven.be	Colin Swaelens KU Leuven colin.swaelens @student.ku- leuven.be	Toon Van Hal KU Leuven toon.vanhal @kuleuven.be
--	---	---	---

Abstract

This paper shows the extent to which treebanks of Ancient Greek play a central role in the ongoing Pedalion project at the University of Leuven. Building on diverse treebanks readily available today, the project aims to make progress in the automated parsing of classical and post-classical Greek texts. Rather than developing new technology as such, our project endeavours to make deliberate and methodical use of the technology that already exists, essentially by *combining* and *adapting* both technology and data. This contribution offers a ‘roadmap’ of our project, surveying (a) the existing work on which we can rely, (b) the strategies which we adopt to reach better results in the automated processing of Ancient Greek and (c) the deliverables that have already been realised or are forthcoming.

1 Introduction

Although corpus-based methods are becoming increasingly central to present-day research in historical linguistics, the possibilities for conducting corpus-based linguistic research in Ancient Greek are still restricted, despite a range of recent international research initiatives (see Haug, 2014). Our ongoing project aims to make some progress in the automated language processing of Ancient Greek. It starts from the basic assumption that promising results with wide-ranging applicability can be achieved by relying on the invaluable work already undertaken in a wide range of Ancient Greek dependency treebank projects. The specifics of our approach can be characterised as follows:

- Rather than developing new technology as such, our project endeavours to make deliberate and methodical use of the technology that already exists, essentially by *combining* and *adapting* both technology and data. In doing so, special attention is paid to the specifics of the Greek language.
- Instead of aiming solely at reaching better parsing accuracy, the project also aims to offer a number of tangible deliverables.
- Such deliverables should not be limited to specialised instruments tailored to the needs of researchers and linguists: there are also didactic applications in development that can assist a larger audience in mastering Ancient Greek.

In what follows, we offer a ‘roadmap’ of our project by succinctly outlining (a) the work on which we are gratefully building (section 2), (b) the strategies we adopt to achieve better results in the automated processing of Ancient Greek (section 3) and (c) the deliverables that have already been realised or are in progress (section 4). In a recent paper, Simon Mahony highlighted the importance of “joining together and sharing resources”, particularly “[i]n the case of ancient languages, just as with other vulnerable subject areas” (Mahony, 2016, 44). We hope that our ongoing project in some measure meets Mahony’s concern.

2 Elaborating on existing treebanks

Two projects are well-known and are prominently present in the yearly CONLL shared task (Zeman et al., 2018), the Perseus Ancient Greek and Latin Dependency Treebanks and the PROIEL Treebank. But there are many additional undertakings that deserve special mention. Hence, we offer a succinct survey

of dependency treebanks of Ancient Greek (for a survey which includes constituency treebanks as well, see Robie, 2017):

- Perseus *Ancient Greek Dependency Treebanks* (AGDT); ca. 550K tokens. Encompasses Archaic poetry, Classical poetry and prose. Offers lemma, morphological, syntactic and (in a few cases) semantic information. Own annotation style. See (Bamman and Crane, 2011).
- PROIEL treebanks; ca. 248K tokens. Encompasses prose texts. Offers lemma, morphological, syntactic and pragmatic information. Own annotation style. See (Haug and Jøhndal, 2008).
- Sematia; ca. 6K tokens. Documentary papyri. Offers lemma, morphological and syntactic information, following the AGDT annotation scheme (with some minor modifications). See (Henriksson and Vierros, 2017).
- Gorman treebanks; ca. 240K tokens. Encompasses prose texts. Offers lemma, morphological and syntactic information. Complies with the AGDT annotation scheme. See (Gorman, 2016).
- Harrington treebanks; ca. 18K tokens. Encompasses prose texts. Offers lemma, morphological, syntactic and semantic information, following a modified version of the AGDT annotation scheme. See (Harrington, 2018).
- Pedalion treebanks; ca. 119K tokens. Offers lemma, morphological and syntactic information, currently experimenting with semantic information, following the AGDT annotation scheme. See below.
- Aphthonius; ca. 7K tokens. Encompasses prose texts. Offers lemma, morphological, syntactic and semantic information, following the AGDT annotation scheme. See (Yordanova, 2018).

In order to be able to join forces with the data sets outlined above and to enable communication between them, we imported the XML-files into a relational FileMaker Database, which serves as the back-office of our undertaking. The annotation styles of both the PROIEL treebank—whose set of syntactic labels is slightly more extensive than the set used in the Perseus treebanks, given that, for instance, special labels are for instance assigned to ‘agent’ and indirect objects (see Haug, 2010)—and the Harrington treebank were automatically converted to the Perseus standards on the basis of a rule-based method.

3 NLP technology and strategies used

The project’s current focus lies on making progress in automated syntactic analysis. Scholars active in the field of stochastic natural language processing approaches to Ancient Greek have so far focused mainly on morphological analysis (see, for instance, Dik, 2018 and Celano et al., 2016). Keersmaekers (2019) recently succeeded in obtaining very promising results for morphology (ca. 95% accuracy) based on a text corpus focusing on the Greek papyri, while also including tokenisation and lemmatisation (the latter with about 99% accuracy) in a pipeline model. This offered a good starting point for further progress in automated syntactic analysis.

Due to the free constituent order and the highly inflected nature of Ancient Greek, progress in automatically analysing Ancient Greek texts is rather slow. Techniques successfully applied to English texts do not guarantee the same level of performance when applied to an Ancient Greek corpus. Lee et al. (2011) achieve an Unlabelled Attachment Score of 70.5% with a joint tagging/parsing model, while the highest Labelled Attachment Score (LAS) Mambrini and Passarotti (2012) report is 71.7%, trained and tested on Homeric Greek. In the most recent CONLL shared task on multilingual syntactic parsing, the highest achieved LAS (with the HIT-SCIR parsing system) is 79.4% for the Perseus treebanks and 79.3% for the PROIEL treebanks (Zeman et al., 2018).

In order to achieve better results in the automatic analysis of Greek sentences, we have developed multiple strategies. Considerations of space prevent us from fleshing out the strategies which have so far been implemented in order to obtain better results (a more extensive overview of the strategies implemented is in preparation). We will limit ourselves to a succinct survey:

- **Expanding** the training data. The fact that the results of machine learning strongly depend on the extent of the available data is sometimes substantiated by referring to a quote by Peter Norvig, Google’s director of Research, who once said that his company did not have “better algorithms, we just have more data” (see, e.g., Rosenfeld and Kraus, 2018: 41). We will discuss this in section 4.1.

- **Homogenising** the training data. It is not only the extent of the data that matters, quality is also key (see, e.g. Schluter and Van Genabith, 2007). An important goal of our research is to make the existing treebank data available more homogeneous, since the number of different annotators and standards has led to a large number of inconsistencies. This will improve the ‘learnability’ of the data for a syntactic parser, as well as create a better standard against which the test data can be evaluated (while also enhancing the possibilities for corpus linguistic research). The complex FileMaker database, containing all tokens of all available Ancient Greek dependency treebanks, has proven to be an invaluable tool in detecting inconsistencies. See section 4.2 for more information.
- **Adapting** the annotation format: the annotation style of the Perseus treebanks is inspired by the one used by the Prague Dependency Treebank and is easily human-readable. However, this does not guarantee that it is also easy to learn for an automatic parsing system. Therefore, we tested which annotation styles are the easiest to learn for specific structures which the parser typically struggles with, including elliptic and coordination structures. We did so by automatically transforming the trees on the basis of a number of rules and testing the accuracy on a test set. For coordination structures, for instance, we found that it is possible to increase parsing accuracy by 5-6% points overall (and 25-30% points for the nodes involved in these structures) if the data are presented in the right format—this involved attaching nodes involved in a coordination structure directly to one of the previous coordinated nodes with the generation relation ‘CO’ (coordinate), in a way comparable to the style of annotation of the Universal Dependencies project.
- **Enriching** the annotation format: we experimented with several features, including enriched part-of-speech tags and semantic information, to further improve parsing accuracy (see also section 4.4).
- **Testing** different parsers (see Mercelis, 2019): our earliest parsing experiments all made use of MaltParser (Nivre et al., 2007). In addition, the integration of MaltOptimizer (Ballesteros and Nivre, 2012) allowed the parser to select the most optimal features for the analysis of Ancient Greek. However, since the results of MaltParser were relatively modest (a LAS of about 0.734 on our test data, cf. section 4.1), we also tested some more recently developed parsers, which use neural networks, i.e. ComboParser (Rybak and Wroblewska, 2018) and the Turku Neural Parser (Kanerva et al., 2018). With the latter in particular we were able to make major improvements, reaching an LAS of up to 90 per cent. However, this number is based on manually annotated morphology, while the numbers are probably lower for automatically morphologically annotated texts.

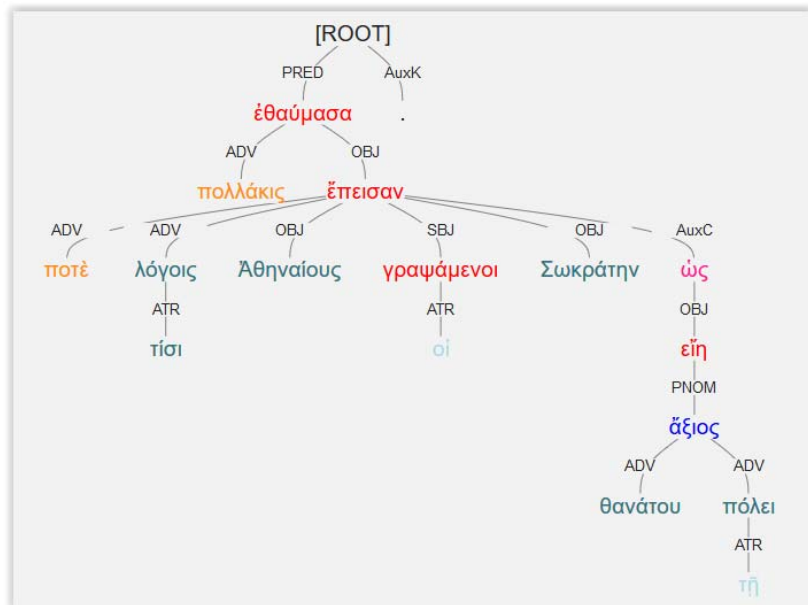


Figure 1: Example of an automatically annotated sentence
 [“πολλάκις ἔθαύμασα τίσι ποτὲ λόγοις Ἀθηναίους
 ἔπεισαν οἱ γραψάμενοι Σωκράτην ὡς ἄξιός εἶη θανάτου τῆ πόλει”]

Nevertheless, it is still relatively difficult to evaluate and compare parsing results, because the data we use still contains several inconsistencies (see above)—let alone the fact that in some cases multiple analyses of the Greek can be defended. Moreover, in many cases having the right head is much more important than having the correct relation (e.g. the distinction between argument and adjunct, which is often fluid), whereas in other cases the reverse is true (e.g. particle attachment). Therefore we combine an automatic evaluation with a close reading of fresh, pre-parsed texts, allowing us to assess the strengths and weaknesses of a new model from a frog’s eye perspective. Figure 1 shows a representative example of an automatically annotated sentence (viz. the first sentence of Xenophon’s *Memorabilia*), with relatively good results. We are also planning to develop new ways to evaluate the test data and the improvement in parsing accuracy in a more detailed manner.

4 Output and applications

4.1 Creating treebanks

Instead of solely aiming to achieve better parsing accuracy, we specifically wanted to offer some tangible deliverables. This is why we extended the rather limited set of morphologically and syntactically annotated prose texts currently available in quantitative terms as well as in terms of its genre diversity, thus significantly increasing the quantity and quality of the training data. By making new trees (see Table 1 for an overview), developed in keeping with the guidelines of the Perseus Dependency Treebanks, we were able to make a detailed analysis of the strengths and weaknesses of the subsequent versions of the Ancient Greek parser under development. The detection of enduring shortcomings and parsing problems also reveals which issues should be prioritised in order to obtain better results. In addition, it allows us to uncover some inconsistencies present in the existing treebanks. The Leuven treebanks were not built from scratch, but on the basis of a pre-tagged and pre-parsed version, which considerably improved and accelerated the treebanking process. Part of these trees (all of which are beta versions) have served as test data in comparing the different parsers (Mercelis, 2019: see *supra*).

Author	Details	Prose/Poetry	#Tokens
Aesop	Select fables	Prose	7,5K
Anon.	Batrachomyomachia	Poetry	2,2K
Aristophanes	Thesmophoriazusae	Poetry	9K
Diverse authors	Papyrus texts	Prose	12K
Diverse authors	Pedalion example sentences	Prose & Poetry	20K
Euripides	Medea	Poetry	10K
Lucian	Prometheus, Symposion, Lis vocalium, Philopseudes 33-36, The Mule	Prose	21K
Lyric Poetry	Mimnermus, Theocritus, Semonides	Poetry	1,5K
Lysias	On the Pension (Or. 24)	Prose	1,5K
Menander	Dyskolos	Poetry	8K
Paeanius	Breviarium (parts of chapter 1)	Prose	6K
Prose authors	Longus, Isocrates, Hippocrates (Fragments)	Prose	1,5K
(Pseudo-)Plato	Cleitophon and Crito	Prose	5,8K
(Septuagint)	Parts of Genesis [For the part-of-speech annotation, we made thankful use of (Kraft, 1988).]	Prose	14K
Total			119K

Table 1: Overview of the recently produced treebanks (with approximate numbers of tokens).

We make use of the very user-friendly open-source Arethusa treebank editor, which is an intuitive tool for building and reviewing treebanks (see Figure 1 for an example). In future versions, we will have to pay particular attention to the metadata of our trees, which are currently rather poor.

Apart from offering manually checked treebanks, our project also encompasses automatically parsed data of ca. 37 million tokens. Given that these data could, despite all the errors inherent to the process, be of immediate interest to linguists of Ancient Greek and represent a syntactic ‘sister’ to *Perseus under*

PhiloLogic (see Dik, 2018), we will make the majority of these data available (copyright issues related to a number of texts included prevent us from publishing the corpus in its entirety). In this stage, we are happy to provide future annotators with pre-parsed versions of specific texts, so as to speed up the annotation process.

4.2 Correcting and modifying treebanks

By creating new treebanks on the one hand and by systematically assembling the data of existing treebanks on the other, we were able to trace inconsistencies and errors in existing treebanks of Ancient Greek. A survey of these modifications is published on our GitHub page, where the Readme file offers more information (<http://github.com/pedalion/treebanks>). The modifications are of various kinds. The number of what we believe are clear errors represent only a minor—although not unsubstantial—part of the file: most suggestions are made for purposes of homogenisation. As it is a work in progress, it is safe to say that this file might also contain a number of changes for the worse. The current release version contains modifications of ca. 120K tokens. These modifications have already been implemented in our own treebank query tool, DendroSearch, of which the functionality is outlined in the following section.

4.3 Querying treebanks: DendroSearch

Despite the abundance of treebank initiatives today, there are hardly any tools available which enable users to perform detailed queries in the treebanks. The Iliados tool (briefly mentioned in Mahony, 2016: 42) is restricted to a relatively small selection of poetic texts in the Perseus’ Ancient Greek Dependency Treebanks. Annis, a tool that can query the Perseus Latin and Ancient Greek Treebank, has been offline since 2013, but recently a graph-based version was developed (see Krause, 2019). The PROIEL treebank can be queried through the INESS-tool (Rosén et al., 2012).

To encourage corpus-based research in the existing treebanks we developed DendroSearch, a stand-alone tool that is explicitly designed to query Greek treebanks in a user-friendly way. Through a series of panels, users can build complex queries and send them to a search system which goes through all available treebank material and presents the results (see Figure 2). For this tool we integrated all the corrections we made, all the conversions between annotation formats we implemented, as well as the treebanks that were produced by our research group, into the existing treebanks. We hope that the tool as well as the source code, which will be made available on GitHub, will be useful to other researchers currently developing treebank query initiatives. In future versions, visualisation capabilities could be improved so as to make querying the treebanks even more intuitive, and a number of basic statistical analytics (e.g. collocation and collostructional analysis) could be introduced. Additionally, a new version will encompass the possibility of performing semantic queries, which is the topic of the next section.

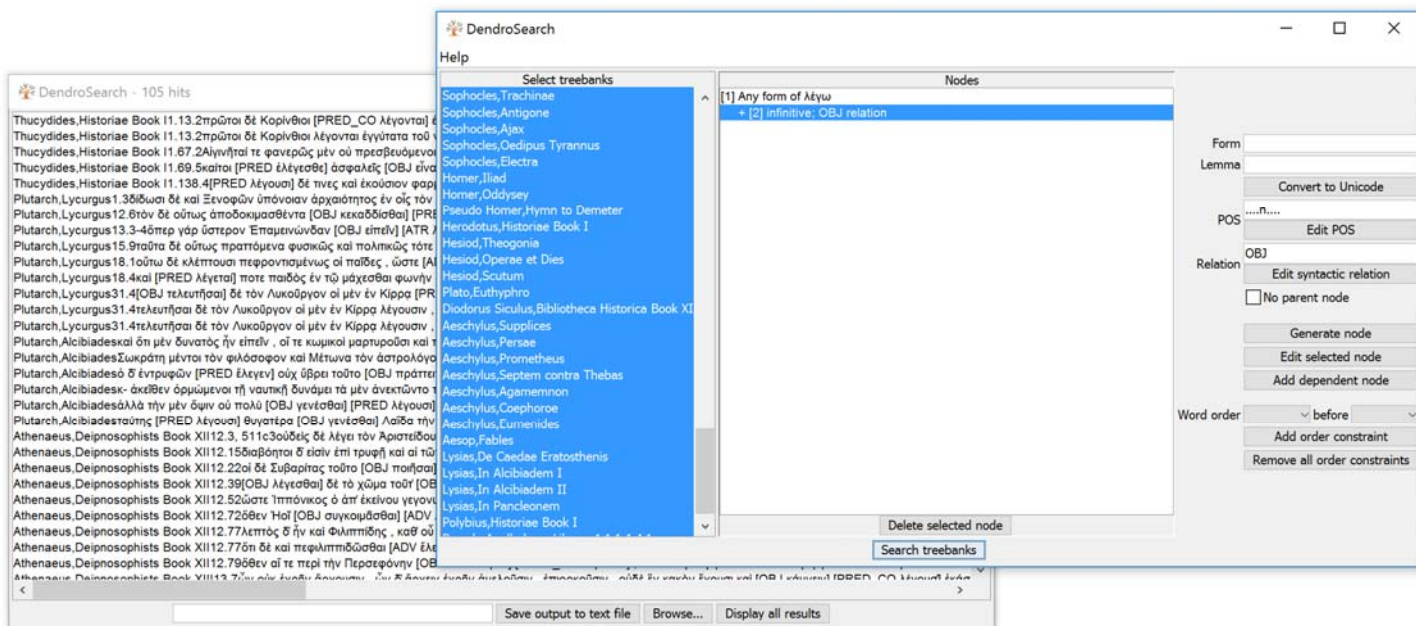


Figure 2: Screenshot of the DendroSearch stand-alone tool

4.4 Enriching treebanks: the role of semantics

We are currently experimenting with adding a semantic layer to the morphological and syntactic annotation (see Swaelens, 2019). The approach is twofold. On the one hand, we aim to assign a semantic hypernym to the lemmas of each noun (e.g. ‘person’, ‘animal’, ‘non-concrete’ etc.), verb (e.g. ‘emotion’, ‘perception’, ‘stative’ etc.) and adjective (‘quantifier’ vs. ‘qualifier’). It is likely that this will further improve parsing accuracy (as some tests have indicated) and also enhance searching possibilities. In addition, we are also experimenting with distributional vectors of Greek lemmas, based on a large automatically annotated corpus of approximately 37 million tokens (see 4.1).

Alongside this lemma-based approach, we try to define the semantic role of adverbials and attributes. The underlying hypothesis is that for certain parts of a sentence the semantic role (e.g. standard of comparison, agent, possessor, direction, etc.) is more significant than the syntactic function (it is often very difficult to make a consistent distinction between adverbials and objects, for instance). Most of the semantic roles were added by student annotators, but we are also developing approaches which will do this automatically or semi-automatically. Semantic role labelling is present in a number of treebank initiatives (viz. the Perseus Ancient Greek Dependency Treebanks, see Celano and Crane, 2015, and the Harrington treebanks). Table 2 displays the semantic roles currently distinguished in the Pedalion project (mainly based on Crespo et al., 2003). Swaelens (2019: 32-34) includes a comparative table contrasting the use of semantic roles in the different Greek Treebank initiatives.

AGENT	DURATION	LOCATION	RESULT
BENEFICIARY	EXPERIENCER	MANNER	SOURCE
CAUSE	EXPLANATION	MATERIAL	TIME
COMPANION	EXTENT OF SPACE	PATIENT	TIME FRAME
CONCESSION	GOAL	POSSESSOR	TOTALITY
CONDITION	IDENTITY	PROPERTY	VALUE/PRICE
DEGREE/MEASURE	INSTRUMENT	RECIPIENT	
DIRECTION	INTERMEDIARY	RESPECT	

Table 2: Overview of the semantic roles currently distinguished in Pedalion.

4.5 Valorising treebanks work in a didactic context

In some recent papers the pedagogical value of making Ancient Greek and Latin treebanks has been highlighted (see e.g. Mambrini, 2016). So far, the focus has been on the educational benefits of treebank *creation*. Annotating a treebank implies close reading and making detailed morphological and syntactic analyses, which will considerably increase a student’s awareness of the complexities and difficulties inherent in Ancient Greek syntax. While subscribing to this view, we also argue that treebanked texts can, and should, play a significant role as *products* and *tools* for receptive language learners as well. Our ongoing project implements three ways of valorising existing treebanks for educational purposes.

Needless to say, a first obvious application consists in offering reading support for treebanked texts or text fragments. The Perseids and Arethusa initiatives, already mentioned above, enable users to create treebanks with beautifully visualised trees of analysed sentences (see Figure 1 for an example). Through collaboration with Perseids and Arethusa staff members we were able to make use of their recently generated “Treebank Template” (<https://github.com/perseids-publications/treebank-template>), which also allows our users to browse through the trees in a convenient and user-friendly way (<http://en.pedalion.org/reading>).

A second application concerns vocabulary. *Chilia*, building on a frequency-based vocabulary of Ancient Greek (Van Hal, 2013), contains the 1000 most frequent lemmas found in Classical Ancient Greek texts. To some extent, its development be seen against the precarious backdrop of Ancient Greek studies in high schools in the Low Countries. Although Ancient Greek is still relatively well-represented in gymnasia programs in Flanders and the Netherlands, the number of pupils is too limited to attract much attention on the part of educational publishers. This explains why teachers are forced to make use of somewhat dated learning tools, which might contribute to a further decline of pupil numbers. On the other hand, this situation also creates the possibility to take the lead in creating Open Educational Resources tailored to the needs of high school pupils. *Chilia* is conceived of as a modest contribution in this direction. The novelty of *Chilia* consists in the fact that every single entry is accompanied by a short

real-life sentence (some of which are slightly abridged) which contextualises the lemma in question. Furthermore, all the sentences included contain only words which occur in the *Chilia* word list (with the exception of proper names). So, for instance, the lemma *pote* is accompanied by the following example sentence from the Athenian orator Andocides (c. 440–c. 390 BC): “ἦν γάρ ποτε χρόνος, ὃ Ἀθηναῖοι, ὅτε τεῖχη καὶ ναῦς οὐκ ἔκεκτήμεθα” [“**Once** there was a time, Athenians, when we had neither walls nor a fleet”]. Given that the other words in this sentence belong to the 1000 most frequent words as well, learners of Ancient Greek are enabled to study vocabulary in context and in a self-reinforcing way. Most sentences were selected by relying on Ancient Greek treebanks that exist today or by specifically searching—in a semi-automatic way—for sentences that meet the required conditions. *Chilia* will be published both as a stable e-publication (which can be downloaded in pdf-format) and in a dynamic online-environment, which will enable users to visualise the syntactic trees of the example sentences and to establish links to other online initiatives, such as *Logeion* (see Dik, 2019).

Treebanks will also play a role as an enhancement of an already existing Open Educational Resource, viz. the online modular grammar of Ancient Greek (Van Hal and Anné, 2017), the English version of which is still partly under construction. This grammar aims to overcome the static nature of traditional grammars by granting users the possibility to switch from the language’s formal level to its semantic, syntactic, or pragmatic level and vice versa through principles of faceted search. The syntax encompasses a large number of original example sentences (many of which stem from post-classical authors, active in the Hellenistic or Roman period), the majority of which have been treebanked. By clicking on a specific example sentence, users can consult the syntactic tree.

5 Conclusions

This paper has presented ongoing work for the Pedalion-project at the University of Leuven. Reasons of space have prevented us from fully substantiating our methods and strategies, but we plan to do so in following publications. An important pillar of this work is the fruitful combination of several existing resources in order to (a) create new linguistically annotated data, (b) improve the quality of the existing data, (c) make the existing data easier to query for users with limited programming skills, (d) expand on the existing data and (e) valorise the data for pedagogical purposes. As for (a), we make use of state-of-the-art NLP technology to quickly create large amounts of new data. The advantage of this strategy is that it is much faster to correct pre-tagged, pre-lemmatised and pre-parsed data than it is to create new treebanks from scratch. As for (b), the homogenisation of existing projects has numerous benefits, including improving the performance of the NLP technology and making it easier to compare its results, as well as making it easier to query these projects for linguistic information and to do so more reliably. As for (c), we have created a user-friendly tool to query the treebanks, DendroSearch, which will allow a broader audience to make use of the various research possibilities that the existing treebanks are already offering. As for (d), we have shown how we plan to add semantic information (at the lemma level as well as in terms of semantic roles) as a valuable supplementary layer for linguistic enquiries. Finally, as for (e), we have created and will continue creating a set of tangible deliverables with pedagogical purposes. As researchers involved in a project that gratefully makes use of the painstaking work done by other people in the scientific community, we also present this paper as a call to invite others to expand on our work (which will be made publicly available on GitHub) as well as discuss new future possibilities of collaboration.

Acknowledgements

We make grateful use of the large number of treebanks readily available. Our treebank data was created and edited with the help of the Arethusa application (<https://github.com/alpheios-project/arethusa>) as provided by the Perseids Project at Tufts University (<https://perseids.org>). (Arethusa is now being jointly maintained by the Perseids Project and The Alpheios Project, Ltd.) Since January 2019, our work is also partly funded through an FWO research grant (Research Foundation Flanders). We are especially indebted to the students who enthusiastically participated in our treebanking classes. Numerous colleagues were willing to assist us in our experiments, to comment on earlier drafts of this paper or to exchange ideas prior to the writing of this short paper. We would therefore like to thank Bridget Almas, Giuseppe Celano, Harry Diakoff, Zach Fletcher, Bob & Vanessa Gorman, Dag Haug, Francesco Mambrini, Merisa Martinez, Reuben Pitts, James Tauber, Demmy Verbeke, Marja Vierros and two anonymous reviewers.

References

- Miguel Ballesteros and Joakim Nivre. 2012. “MaltOptimizer: An optimization tool for MaltParser.” In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 58–62. Association for Computational Linguistics, Stroudsburg. <<https://www.aclweb.org/anthology/E12-2>>
- David Bamman and Gregory Crane. 2011. “The Ancient Greek and Latin Dependency Treebanks.” In *Language Technology for Cultural Heritage*, edited by Caroline Sporleder, Antal van den Bosch and Kalliopi Zervanou, 79–98. Springer, Berlin & Heidelberg.
- Giuseppe G. A. Celano and Gregory Crane. 2015. “Semantic Role Annotation in the Ancient Greek Dependency Treebank.” In *Proceedings of the Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT14)*, edited by Markus Dickinson et al., 26–34. Institute of Computer Science, Warsaw. <http://tlt14.ipipan.waw.pl/files/4614/5063/3858/TLT14_proceedings.pdf>
- Giuseppe G. A. Celano, Gregory Crane, and Saeed Majidi. 2016. “Part of Speech Tagging for Ancient Greek.” *Open Linguistics* 2(1):393–399. doi:10.1515/opli-2016-0020.
- Emilio Crespo, Luz Conti, and Helena Maquieira. 2003. *Sintaxis del griego clásico*. Gredos, Madrid.
- Helma Dik. 2018. *Perseus under PhiloLogic*. <<http://perseus.uchicago.edu/>>
- Helma Dik. 2019. *Logeion*. <<https://logeion.uchicago.edu/>>
- Vanessa Gorman. 2016. *Gorman Treebanks*. <<https://history.unl.edu/vanessa-b-gorman>> and <https://github.com/rgorman/author_attribution/tree/master/vg_combined_trees2>
- Matthew Harrington. 2018. *Perseids Project - Treebanked Commentaries at Tufts University*. <https://perseids-project.github.io/harrington_trees/>
- Dag T. T. Haug. 2010. “PROIEL Guidelines for Annotation” <http://folk.uio.no/daghaug/syntactic_guidelines.pdf>.
- Dag T. T. Haug. 2014. “Computational Linguistics and Greek.” In *Encyclopedia of Ancient Greek Language and Linguistics*, edited by Georgios K. Giannakis, 1:354–356. Brill, Leiden & Boston.
- Dag T. T. Haug and Marius Jøhndal. 2008. “Creating a Parallel Treebank of the Old Indo-European Bible Translations.” In *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, edited by Caroline Sporleder and Kiril Ribarov, 27–34. <<https://www.hf.uio.no/ifi-ikk/english/research/projects/proiel/Activities/proiel/publications/marrakech.pdf>>
- Erik Henriksson and Marja Verros. 2017. “Preprocessing Greek Papyri for Linguistic Annotation.” *Journal of Data Mining & Digital Humanities. Special Issue on Computer-Aided Processing of Intertextuality in Ancient Languages*. <<https://hal.archives-ouvertes.fr/hal-01279493v1/document>>
- Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. “Turku Neural Parser Pipeline: An End-to-End System for the CoNLL 2018 Shared Task.” In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, 133–142. Association for Computational Linguistics, Brussels. <<http://www.aclweb.org/anthology/K18-2013>>
- Alek Keersmaekers. 2019. “Creating a Richly Annotated Corpus of Papyrological Greek: The possibilities of natural language processing approaches to a highly inflected historical language.” *Digital Scholarship in the Humanities* (online): 1–16. doi:10.1093/lc/fqz004.
- R. Kraft. 1988. *Morphologically Analyzed Septuagint (version 1.0)*. Computer-Assisted Tools for Septuagint Studies (CATSS). University of Pennsylvania. <<http://ccat.sas.upenn.edu/gopher/>>
- Thomas Krause. 2019. *ANNIS: A graph-based query system for deeply annotated text corpora*. PhD-Dissertation Humboldt Universität zu Berlin. <<https://edoc.hu-berlin.de/handle/18452/20436>>.
- John Lee, Jason Naradowsky, and David A. Smith. 2011. “A Discriminative Model for Joint Morphological Disambiguation and Dependency Parsing.” In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human language technologies*, 1:885–894. Association for Computational Linguistics, Stroudsburg. <<https://www.aclweb.org/anthology/P11-1089>>
- Simon Mahony. 2016. “Open Education and Open Educational Resources for the Teaching of Classics in the UK.” In *Digital Classics Outside the Echo-Chamber: Teaching, knowledge exchange & public engagement*, edited by Gabriel Bodard and Matteo Romanello, 33–50. Ubiquity Press, London. <<https://oopen.org/search?identifier=649985>>
- Francesco Mambrini. 2016. “The Ancient Greek Dependency Treebank: Linguistic annotation in a teaching environment.” In *Digital Classics Outside the Echo-Chamber: Teaching, knowledge exchange & public engagement*, edited by Gabriel Bodard and Matteo Romanello, 83–99. Ubiquity Press, London. doi:10.5334/bat.f
- Francesco Mambrini and Carlo Passarotti. 2012. “Will a Parser Overtake Achilles? First experiments on parsing the Ancient Greek Dependency Treebank.” In *Eleventh International Workshop on Treebanks and Linguistic Theories*, 133–144. Edições Colibri. <<https://pdfs.semanticscholar.org/b5b5/4154385697b29fe3d9f0fce080c7d34525cd.pdf>>

- Wouter Mercelis. 2019. *Syntactisch parsen van Oudgriekse teksten: een vergelijkende studie*. Unpublished master thesis KU Leuven.
- Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. “MaltParser: A language-independent system for data-driven dependency parsing.” *Natural Language Engineering* 13(2):95–135. doi:10.1017/S1351324906004505
- Jonathan Robie. 2017. “Nine Kinds of Ancient Greek Treebanks.” *Open Data for Digital Biblical Humanities*. <<http://jonathanrobie.biblicalhumanities.org/blog/2017/12/20/treebanks-for-ancient-greek/>>
- Victoria Rosén, Koenraad De Smedt, Paul Meurer, and Helge Dyvik. 2012. “An open infrastructure for advanced treebanking.” In *META-RESEARCH Workshop on Advanced Treebanking at LREC2012, Istanbul, Turkey, May 2012*, edited by Jan Hajič, Koenraad De Smedt, Marko Tadić, and António Branco, 22–29. European Language Resources Association (ELRA), Istanbul. <<https://ling.w.uib.no/files/2013/02/lcrec2012at-iness-paper-published.pdf>>
- Ariel Rosenfeld and Sarit Kraus. 2018. *Predicting Human Decision-making: From prediction to action*. Morgan & Claypool, s.l.
- Piotr Rybak and Alina Wroblewska. 2018. “Semi-Supervised Neural System for Tagging, Parsing and Lemmatization.” *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*: 45–54. Brussels: Association for Computational Linguistics, Brussels. <<http://www.aclweb.org/anthology/K18-2004>>
- Natalie Schluter and Josef van Genabith. 2007. “Preparing, Restructuring, and Augmenting a French Treebank: Lexicalised parsers or coherent treebanks?” In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, 200–209. <http://doras.dcu.ie/15265/1/78_Paper_meta.pdf>
- Colin Swaelens. 2019. *De rol van semantiek bij de automatische zinsanalyse van het Grieks*. Unpublished master thesis KU Leuven.
- Toon Van Hal. 2013. *Ankura. Basiswoordenlijst Oudgrieks*. Garant, Antwerpen & Apeldoorn.
- Toon Van Hal and Yannick Anné. 2017. “Reconciling the Dynamics of Language with a Grammar Handbook. On Pedalion, an ongoing Greek grammar project.” *Digital Scholarship in the Humanities* 32(2):448–454.
- Polina Yordanova. 2018. *Treebank of Aphonius, Progymnasmata*. <<https://github.com/polinayordanova/Treebank-of-Aphonius-Progymnasmata>>
- 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*, 1–21. Association for Computational Linguistics, Brussels. <<https://www.aclweb.org/anthology/K18-2001>>

Syntax is clearer on the other side – Using parallel corpus to extract monolingual data

Andrea Dömötör

Pázmány Péter Catholic University / H-1088 Budapest, Szentkirályi str. 28.
MTA-PPKE Natural Language Processing Group / H-1083 Budapest, Práter str. 50/a.
domotor.andrea@itk.ppke.hu

Abstract

This paper describes the elaboration of a training corpus containing Hungarian sentences that are labelled according to a syntactic criterion, namely the syntactic role of a very common multifunctional word *volt* 'was/had'. The labels are assigned by a rule-based algorithm that specifies the function of the target word based on the English pairs of the sentences extracted from a parallel corpus. The reasoning of this idea is that the required syntactic information is easier to retrieve in English than in Hungarian. The accuracy achieved by the algorithm was fair but still needs improvement in order to use the output as reliable training data. The obtained training corpus was tested with FastText's text classifier, the results of which showed that the targeted disambiguation problem is resolvable using neural network based text classification.

1 Introduction

In the past years deep learning methods have come to dominate in most of the areas of computational linguistics. A general advantage of these is their robustness and relative simplicity compared to rule-based systems. The key of success in deep learning is having a large and good set of training data, therefore corpus building has become an important field of research.

This paper describes the elaboration of a training corpus containing Hungarian sentences that are labelled according to a syntactic criterion, namely the syntactic role of a very common multifunctional word *volt* 'was/had'. The labels are assigned by a rule-based algorithm that specifies the function of the target word based on the English pairs of the sentences extracted from a parallel corpus. The reasoning of this idea is that the required syntactic information is easier to retrieve in English than in Hungarian.

1.1 The deep learning task

The targeted deep learning task is a word sense disambiguation problem in Hungarian, namely the automatic handling of the multifunctionality of the word *volt* 'was/had'. This token can either be a lexical verb used in locative and possessive sentences (Examples 1, 2) or a copula in case of nominal predicates (Example 3).

- | | | | |
|--------|---|--------|--|
| (1) a. | <i>Ádám otthon volt.</i>
Adam at_home be-PST-Sg3
'Adam was at home' | b. | <i>Van egy macskám.</i>
have-Sg1 a cat-Poss.Sg1
'I have a cat.' |
| b. | <i>Ádám otthon van.</i>
Adam at_home be-Sg3
'Adam is at home' | (3) a. | <i>Éva nagyon szerény volt.</i>
Eve very humble AUX-PST-Sg3
'Eve was very humble.' |
| (2) a. | <i>Volt egy macskám.</i>
have-PST-Sg1 a cat-Poss.Sg1
'I had a cat.' | b. | <i>Éva nagyon szerény.</i>
Eve very humble
'Eve is very humble.' |

The main difference between these functions is that *volt* in Example 3 is omitted in present tense 3rd person while the locative and possessive verbs (Examples 1 and 2) have their present forms *van*. Based on this characteristic of the examined sentence types, this research aims to differentiate between two functions of the word *volt*. These functions will be referred as *copula* (Example 3) and *lexical verb* (Examples 1 and 2) later on. These denominations are different from the Anglo-Saxon terminology where the locative *be* is also considered a copula. However, the studies on Hungarian syntax often narrow the meaning of copula to the auxiliary verb of the nominal predicate because of its exclusive capability of having a zero form. This study follows this traditional Hungarian terminology for the same reason.

In dependency parsing a lexical verb should be considered the head of the sentence while the copula (which can be omitted at least in some persons or tenses) is a complement of the predicative nominal, according to the annotation guidelines of Universal Dependencies (Nivre, 2014). Therefore, the disambiguation of these functions is crucial for parsing. However, as seen in Examples 1 and 3, disambiguation cannot be made based on corresponding lexical items (*be* or *have*) alone because the verb of locative sentences and the auxiliary of the nominal predicate also need to be distinguished, and these are both represented by *be* in English. The disambiguation of the functions of *be* requires a deeper analysis of parse structures.

1.2 The aimed solution

Copular, locative and possessive sentences have clear distinctive structural characteristics, however, a rule-based method is not effective for Hungarian. One source of difficulty is that in Hungarian the word order does not define the syntactic role of the words. Other characteristic that complicates the automatic handling of Hungarian is that it is a so-called pro-drop language, which means that the subject of the sentence is not necessarily overt. Both mentioned characteristics of Hungarian syntax obstacle the detection of predicative nominals to such an extent that the specification of the sentence types listed above would need an in-depth analysis (morphology and NP-chunking). It seems more advantageous to solve this problem with a deep learning method, like neural network based sentence classification.

For this approach a large amount of labelled data is required. This study focuses on the acquisition of training data for a sentence classifier. The obtained data was tested with FastText’s text classifier (Joulin et al. (2016), Bojanowski et al. (2016)).

1.3 Baseline results

The results will be compared to the performance of the e-magyar toolset which is an integrated text processing pipeline for Hungarian (Váradi et al. (2018)). The system has 8 modules that cover the most common NLP tasks (tokenizer, morphological analyzer, lemmatizer, POS tagger, dependency parser, constituent parser, NP chunker, NER tagger). For the specific task of this paper I used the dependency parser module (which obviously uses the analyses of the modules of lower levels). A test set of 1000 sentences was parsed and classified according to the parser’s analyses. If there was a word in PRED relation with *volt* the sentence was assigned a copular tag, otherwise it received a lexical tag. The tags were reviewed manually. The results are displayed in Table 1.

Erroneous labels	186
Accuracy	81,4%

Table 1: Results of the evaluation of the e-magyar tool on 1000 sentences

As the achieved accuracy result shows, the monolingual pipeline analysis struggles with the ambiguity of *volt*.

2 Method

A neural network based sentence classifier that could solve the problem described in Section 1.1 needs training data with sentences that are annotated with the corresponding function (verb or copula) of the target word. As manual labelling is time-consuming, it was inevitable to find a method for automatic

labelling. The basic idea of this method is to use an English-Hungarian parallel corpus. Contrary to Hungarian, English has a restricted word order and no pro-drop, which characteristics allow to make syntactic decisions based on local information. That means that the English pairs of the Hungarian sentences can help to define the function of the word *volt*, by applying fewer and simpler rules as if we used the Hungarian part only.

2.1 The parallel corpus

For data extraction I used an English-Hungarian lemmatized, morphologically analyzed and disambiguated, word-aligned corpus (Novák et al., 2019). This research did not contribute to the creation of this corpus.

The base of the corpus is OPUS Opensubtitles (Lison and Tiedemann, 2016) which contains 644,5 million tokens of aligned sentences. As first step, both sides of the corpus were morphologically analyzed and disambiguated. The English side was lemmatized with the morpha tool (Minnen et al., 2001) and tagged with Stanford tagger (Toutanova et al., 2003). On the Hungarian side the lemmatization and disambiguation was made with PurePos (Orosz and Novák, 2013) which uses the analyses of the Humor analyzer (Novák, 2014). The analyzed texts were transformed on both sides so that every original token is represented by two tokens: (1) the lemma and its main POS-tag and (2) other morphosyntactic tags belonging to the token.

Example 4 shows a pair of preprocessed sentences (*Szeretlek, kedvesem. – I love you, dear*).

- (4) a. szeret[IGE] [Ie1] ,[PUNCT] kedves[FN] [PSe1][NOM]
 b. I#PRP love#VB [P] you#PRP ,#, dear#RB

The preprocessed sentences were word aligned with the fast align programme (Dyer et al., 2013). The alignments of Example 4 are displayed in Table 2.

szeret[IGE]	love#VB
[Ie1]	I#PRP, you#PRP
,[PUNCT]	,#,
kedves[FN]	dear#RB
[PSe1][NOM]	dear#RB

Table 2: The alignments of Example 4

2.2 The labelling algorithm

Having the prepared parallel corpus, the first step was to extract the sentences that contained a form of the target word (*volt*) on the Hungarian side. These sentences were labelled according to the syntactic role (copula or lexical verb) of the target word with a rule-based algorithm implemented in Python3.

The labelling programme first checks the English tokens aligned to *volt*. If *volt* is aligned to a non-auxiliar *have* or an expletive *there*, the sentence is labelled as lexical. If the target word is aligned to a form of *be*, the sentence can either be copular or locative, therefore further rules are required to make the decision. In other cases, the sentence is dismissed because if none of the above listed tokens is aligned to *volt*, the English pair of the sentence can not be used for labelling reliably.

In case of *volt* aligned to *be*, the algorithm selects a "keyword" on the English side, the Hungarian alignments of which define the label of the sentence. The keyword is supposed to represent a (part of a) nominal predicate or a non-nominative argument. Therefore, the algorithm searches for the canonical position of these in English sentences.

For keyword selection the programme first specifies whether the sentence is interrogative. If the sentence is declarative the keyword is the first token following *be* that is not an NP-modifier (*very, more* etc.) or a word of negation (Example 5). If the sentence is a yes-no question or its question word is *what, who, whose, which, how* or *why*, the programme follows the same principles as with declaratives but skips one more word due to the inversion of word order (Example 6). If the sentence has another question word (*where, when* etc.), the sentence is labelled as lexical.

(5) a. *Régen ez egy minőség volt.*

It used to be **a** quality.

b. *Nem volt otthon.*

He was not **at** home.

(6) a. *Mi volt ez a zaj?*

What was that **noise**?

b. *Miről volt szó?*

What was it **about**?

The algorithm then checks the morphological tags aligned to the keyword and labels the sentence based on these. The sentence is assigned a lexical label if the aligned morphological tag is a non-nominative case marker. If the keyword is aligned to a determiner or a nominative nominal the sentence is labelled copular. The tags listed in Table 3 cover all the morphological tags that are aligned to a keyword in the corpus.

lexical		copula	
HA	adverb	DET	determiner 'the, a an'
HA NM	adverbial pronoun	DET NM	determinative pronoun
NU	nominal postposition	MN	adjective
INE	inessive 'in'	MN NM	adjectival pronoun
SUP	superessive 'on'	FOK	comparative adjective
ELA	elative 'from inside'	FF	superlative adjective
ADE	adessive 'at (place)'	SZN	numeral
ESSMOD	modal essive '-ly'	SZN NM	numeral pronoun
ILL	illative 'into'	FN	noun
ALL	allative 'onto'	FN NM	nominal pronoun
SUB	sublative 'to (somewhere)'	PS	possessive nominal
CAU	causative 'for (reason)'	OKEP	'-ing'
ABL	ablative 'of'	MI	past participle (adjectival)
HIN	past participle (passive constructions)		
INS	instrumental 'with'		
DEL	delative 'about'		
DAT	dative 'to (someone)'		
TER	terminative 'until'		
TEM	temporal 'at (time), during'		
ESSNUM	numeral essive '(three) of us'		

Table 3: The morphological tags aligned to keywords and the assigned labels

The algorithm also applies some special lexical rules where the morphological tags would be misleading. First, we should mention a special construction that Kádár (2011) calls *environmental copula construction*. These are NP + VAN 'be' constructions that comprise weather, ambient or environmental conditions. Environmental copula constructions do not behave as "other" copular constructions: they do not omit the copula in present tense third person. This means they should be labelled as sentences with a lexical verb, but the keyword-based part of the algorithm would obviously tag them as copular (see Example 7).

(7) a. *Sötét volt és köd.*

dark be-PST-Sg3 and fog

'It was dark and foggy.'

b. It was dark and foggy.

Therefore, these constructions are handled lexically, based on a list of nominals that usually form a part of an environmental copular construction.

There are other cases where keyword selection fails and these could be called consistent translational differences. This means that some English copular clauses are consistently translated to Hungarian with a lexical verb.

The most common case of this is the translation of "being right". As seen in Example 8, in Hungarian "being right" is literally expressed as "having the truth" which is, syntactically, a possessive structure but the algorithm labels it as copular based on its English pair. The case of "being lucky" is similar (see Example 9), however, this expression also has a copular version in Hungarian.

(8) a. *Igazad volt.*
truth-Poss.Sg2 have-PST-Sg3

'You were right.'

b. You were right.

(9) a. *Neki volt szerencséje.*
he-DAT have-PST-Sg3 luck-Poss.Sg3

'He had luck.'

b. He was lucky.

The algorithm handles these cases (and two further similar ones: "being necessary" and "being ready") with exceptional lexical rules.

The labelling algorithm is summarized in Table 4.

Step 1: Check aligns of <i>volt</i>	
<i>have</i>	lexical
<i>there</i>	lexical
<i>be</i>	go to Step 2
other	dismiss sentence
Step 2: Special lexical rules	
environmental copular construction	lexical
<i>right, lucky, necessary, ready</i>	lexical
other	go to Step 3
Step 3: Keyword selection	
declarative sentence	token following <i>be</i>
yes-no question	<i>be</i> + 2 tokens
<i>what, who, whose, which, how, why</i>	<i>be</i> + 2 tokens
other wh-question	lexical
Step 4: Assign label according to keyword	

Table 4: Summary of the labelling algorithm

2.3 Sentence classification

The obtained labelled corpus was used as training data for FastText's text classifier. I prepared two versions of the training corpus: one contains the original sentences while in the other the sentences are represented with the POS-tags of their words only. Both corpora were trained for the same classification task.

3 Results

The output of the labelling script was 791130 labelled sentences, 458270 of which was tagged as copular and 332860 as containing a lexical verb. These numbers show that the target word - as expected - is extremely common which allows to build a reasonably big corpus for our specific task.

The performance of the algorithm was evaluated on a random sample of 1000 sentences, 598 of which is copular and 402 contains a lexical *volt*. (The same sentences were used for the baseline test described in Section 1.3.) The labels that the algorithm gave on this sample were reviewed manually, and also corrected so that FastText could use the same sample as gold standard test data. The results are displayed in Table 5.

Erroneous labels	108
Accuracy	89,2%

Table 5: Results of the evaluation of the labelling algorithm on 1000 sentences

The labelling algorithm overperformed the baseline result (81,4%) significantly, however the achieved accuracy is still far from a gold standard training corpus. The obtained labelled corpus was subject to the neural network based classification experiment anyways.

The accuracy results of FastText classifier are displayed in Table 6. As seen, the classifier works well despite the deficiencies of the training corpus.

Original sentences	89,6%
POS-tags	91,5%

Table 6: Results of sentence classification (FastText)

4 Discussion

As seen in Section 3 both the labelling algorithm and the sentence classifier achieved significantly higher accuracy than the baseline, however, the quality of the training corpus still needs to be improved. This section reviews the labelling algorithm's most common reasons of failure and the possibilities to avoid them.

4.1 Translational differences

The error analysis of the labelling algorithm revealed that the major part of errors does not originate from the algorithm itself. There are labelling mistakes that can be considered "extraneous", because they are caused by erroneous POS-tagging or alignment. Other very common sources of errors are the occasional differences between the English sentences and their Hungarian translations. The algorithm attempts to avoid this problem by disregarding those sentences where *volt* is not aligned to either *be* or *have*. But this constraint still allows a considerable number of sentences where the inconsistent structural, or sometimes also semantic differences of the paired sentences cause difficulties to the labelling algorithm. In Example 10 the Hungarian sentence (10a) is copular but in its English pair (10b) the verb (aligned to *volt*) is *have*, therefore the algorithm assigned a lexical label to the sentence. Example (11a) is a locative sentence but the programme considered it copular based on its English version (11b), which is indeed copular.

- | | |
|---|--|
| <p>(10) a. <i>Egy rossz álom volt.</i>
 a bad dream AUX-PST-Sg3
 'It was a bad dream.'
 b. You had a bad dream.</p> | <p>(11) a. <i>Ők voltak itt először.</i>
 they be-PST-Pl3 here first
 'They were here first.'
 b. They were the first ones here.</p> |
|---|--|

These errors can hardly be avoided, however, the handling of translational differences may worth further consideration. Other possible solution could be the use of parallel corpora with "stricter" translations, like documents of the European Union. The disadvantage of this approach would be the limited domain.

4.2 Special cases

The error analysis also revealed some special cases that are not covered properly by the current version of the algorithm.

A recurrent problem was the handling of nominals with arguments, like "being sure about something" or "being responsible for something" (Example 12). In some of these cases the argument is omitted in the English sentence but it is present in its Hungarian pair. Therefore, the case marker of the argument on the Hungarian side is aligned to the English nominal which is often the labelling algorithm's keyword. As described in Section 2.2 a keyword aligned to a non-nominative case marker indicates that the sentence has a lexical *volt* which is not true in these cases.

- (12) a. *bárki is volt érte a felelős.*
whoever ever AUX-PST-Sg3 it-CAU the responsible
'whoever was responsible for it.'
b. whoever was responsible.

The handling of these special cases needs a more detailed analysis.

5 Conclusions

The main idea of this paper was to retrieve syntactic information in a parallel corpus, by relying on another language in which the automatic disambiguation of the structure is easier. The described algorithm uses English sentences to define the syntactic role of a target word in the Hungarian translations. The goal was to create a labelled corpus that can be used as training data for a neural network based sentence classifier.

The results show proof of concept for the idea, although the accuracy still needs to be improved. The classifier, however, seems to deal fairly with the deficiencies of the training corpus, especially if we use the POS-tags instead of words. The cause of the difference of performance of the two kinds of training corpus may be the small size of the corpora. If only the POS-tags are used the vocabulary is significantly smaller which facilitates the creation of good embeddings. The successful classification based on POS-tags also demonstrates that the difference between copular and lexical *volt* is in great part coded in the sentence structure.

In sum, the experiments described in this paper demonstrated that parallel corpora can be useful to support syntactic analysis in any cases where the targeted structure is more explicit in an another language. On the other hand, FastText's results confirmed that neural network based text classifiers are not for sentiment or topic identification only, they can capture structural differences as well.

References

- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*.
- Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648. Association for Computational Linguistics.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *CoRR*.
- Edit Kádár. 2011. Environmental Copula Constructions in Hungarian. *Acta Linguistica Hungarica*, 2011(4):417–447.
- Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).
- Guido Minnen, John A. Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.
- Joakim Nivre. 2014. Nonverbal Predication and Copulas in UD v2. <http://universaldependencies.org/v2/copula.html>. Accessed: 2019-02-27.
- Attila Novák. 2014. A New Form of Humor – Mapping Constraint-Based Computational Morphologies to a Finite-State Representation. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1068–1073, Reykjavik, Iceland, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L14-1207.
- Attila Novák, László János Laki, and Borbála Novák. 2019. Mit hozott édesapám? Döntést – Idiomatikus és félig kompozicionális magyar igei szerkezetek azonosítása párhuzamos korpuszból [identification of Hungarian idiomatic and light verb constructions from a parallel corpus]. In *XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019) [15th Hungarian Conference on Computational Linguistics]*, pages 63–71, Szeged. Szeged University.
- György Orosz and Attila Novák. 2013. PurePos 2.0: a hybrid tool for morphological disambiguation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013)*, pages 539–545, Hissar, Bulgaria. Incoma Ltd. Shoumen, Bulgaria.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tamás Váradi, Eszter Simon, Bálint Sass, Iván Mittelholcz, Attila Novák, Balázs Indig, Richárd Farkas, and Veronika Vincze. 2018. E-magyar – A Digital Language Processing System. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, H el ene Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA).

Improving Surface-syntactic Universal Dependencies (SUD): surface-syntactic relations and deep syntactic features

Kim Gerdes

Almanach (Inria), LPP (CNRS)
Sorbonne Nouvelle
kim@gerdes.fr

Bruno Guillaume

Université de Lorraine, CNRS,
Inria, LORIA, Nancy, France
bruno.guillaume@inria.fr

Sylvain Kahane

Modyco,
Université Paris Nanterre & CNRS
sylvain@kahane.fr

Guy Perrier

Université de Lorraine, CNRS,
Inria, LORIA, Nancy, France
guy.perrier@loria.fr

Abstract

SUD is an annotation scheme for syntactic dependency treebanks, near isomorphic to UD (Universal Dependencies). Contrary to UD, it is based on syntactic criteria (favoring functional heads) and the relations are defined on distributional and functional bases. In this paper, we will recall and specify the general principles underlying SUD, present the updated set of SUD relations, discuss the central question of MWEs, and introduce an orthogonal layer of deep-syntactic features converted from the deep-syntactic part of the UD scheme.

1 Introduction

SUD (Surface-syntactic Universal Dependencies) is an annotation scheme that we proposed in a previous paper (Gerdes et al., 2018) as an alternative of the UD (Universal Dependencies) annotation scheme (Nivre and al., 2019). SUD follows surface syntax criteria (especially distributional criteria) and can be automatically converted into the UD scheme. SUD has now been used in the development of a treebank for Naija (Courtin et al., 2018; Caron et al., 2019) and treebanks for French and Chinese are in development. Some principles underlying SUD have been further clarified and will be exposed here.

Section 2 recalls and specifies the general principles of SUD. For a more detailed explanation of these principles, we refer the reader to the initial SUD presentation (Gerdes et al., 2018). The following sections present the original points of the article. Section 3 presents the set of SUD relations, which has been updated, providing a better distinction between surface syntactic and deep syntactic features (following the separation between surface and deep syntax of the Meaning-Text Theory (Mel'čuk, 1988)). Section 4 discusses the need for a separate encoding for MWEs' POS in SUD. Section 5 presents some principles of the UD \Leftrightarrow SUD conversion.

2 General principles of SUD

2.1 Surface-syntactic criteria for heads

We will briefly recall the criteria for surface-syntactic headedness. These criteria have been the subject of much discussion (Hudson, 1984; Hudson, 1987; Mel'čuk, 1988). In the original paper (Gerdes et al., 2018), we retain two central criteria: First, the surface syntactic head of a unit U is an element of U that determines the distribution of U , that is, the syntactic position that U can occupy; for instance, *Mary* cannot be the head of $U = to\ Mary$, because *Mary* and U occupy completely different syntactic positions.¹ Such a criterion favors functional heads, while UD treats functional elements as leaves and

¹One exception to this is the case wh-words: although *it is perfect* and *which is perfect* have different distributions (the relative clause modifies a noun), we decided not to take the wh-word at the syntactic head, but to favor its pronominal role inside the relative clause. This is not a theoretical choice, but rather a pragmatic decision preserving the tree structure.

poses as a principle that syntactic relations must be between content words, functional words being then relegated to being markers of the content words.

In some cases, the first criterion does not give a clear situation because two words have head features. In this case, a second gradual criterion comes into play where we prefer to give the status of dependent to the one that changes less the distribution of the unit. According to this principle, a coordinative conjunction such as *and* does not govern the conjunct following it, because *and Mary*, *and red*, or *and is sleeping* occupy completely different positions. In the same way, the determiner is analyzed as a dependent of the noun because nouns partly control the distribution of a combination determiner-noun (*this morning* can work as a modifier of a verb contrary to *this boy*).

A last point concerns coordination: SUD adopts a string-analysis of coordination, where each conjunct depends on the previous one, contrary to UD, which adopts a bouquet-analysis, where each conjunct depends on the first conjunct. One of the key arguments for the string-analysis is that it reduces the dependency length (Gibson, 1998; Liu, 2008; Futrell et al., 2015).

2.2 Criteria for SUD relations

SUD relations (that is, dependency labels) are defined by means of functional criteria: Two units that commute in the same syntactic position (and consequently bear the same function) must be linked to their governor by the same relation. The characterization of a relation is based on the whole paradigm of elements that can commute in the dependent position, while UD relations strongly rely on the POS of the dependent. For instance, a unique *comp:obj* relation for direct object complements is considered in SUD, where UD considers three relations: *obj* for a nominal object (*I imagine a dance*), *ccomp* for a clausal object (*I imagine (that) he dances*) and *xcomp* for a clausal object without its own subject (*I imagine to dance*).

This last relation raises another problem. (Przepiórkowski and Patejuk, 2018) extensively argue that UD's *xcomp* is particularly unsatisfactory because it is based on a property (not having its own subject), which is orthogonal to the syntactic function and can even be realized with modifiers (*He came without running*).² We make a clear distinction between surface-syntactic properties, which determine relation classes, and deep-syntactic properties, such as those expressed by *xcomp*. In Section 3.2, we will propose to represent deep-syntactic properties with specific relation extensions.

Hence, a subset of 17 UD relations (*nsubj*, *csubj*, *obj*, *iobj*, *obl*, *xcomp*, *ccomp*, *amod*, *nmod*, *nummod*, *advmod*, *acl*, *advcl*, *aux*, *cop*, *case*, *mark*) is replaced by 3 major relations in SUD: *subj*, *comp*, *mod* (subject, complement, modifier) with possible sub-relations.³

3 SUD relations

SUD relations are organized in a taxonomic hierarchy (Figure 1): A relation that is the daughter of another one inherits its syntactic properties with the addition of specific properties. Indeed, sometimes, we cannot take into account all possible distinctions, either because of the conversion from different treebanks not containing enough information, or because a sentence does not allow to make a clear decision. In those cases, we need a more general class of relations. For example in *They work >udep at the university* out of context does not allow distinguishing between *mod* and *comp:obl*, and we can then use *udep* (underspecified dependency), the hypernym of *mod* and *comp*. The root of our taxonomy is the *unk* (unknown) relation.

Some UD relations are used in SUD with the same scope and meaning as in UD (in the green frame on Figure 1, whereas UD relations that are not used in SUD are listed in the orange frame), except for some cases where UD is particularly restrictive (see Section 3.2). Also, the sets of POS and morpho-syntactic features are similar in SUD and UD.

²Moreover, UD is not consistent about when to distinguish clausal complements and modifiers (*He wants to run* is *xcomp* and *He came without running* is *advcl*), while not making the same distinction for adpositional phrases (*He spoke to her* and *He came without her* are both *obl*).

³The distinction between arguments and modifiers mainly involve a semantic criterion: an argument of a lexical unit **L** is an obligatory participant in the semantic description of **L** (Mel'čuk, 1988). Although semantic, we want to keep this distinction in the syntactic annotation because most languages have special constructions for arguments such as the English dative shift and the French indirect object complement, which can be pronominalized by a dative clitic.

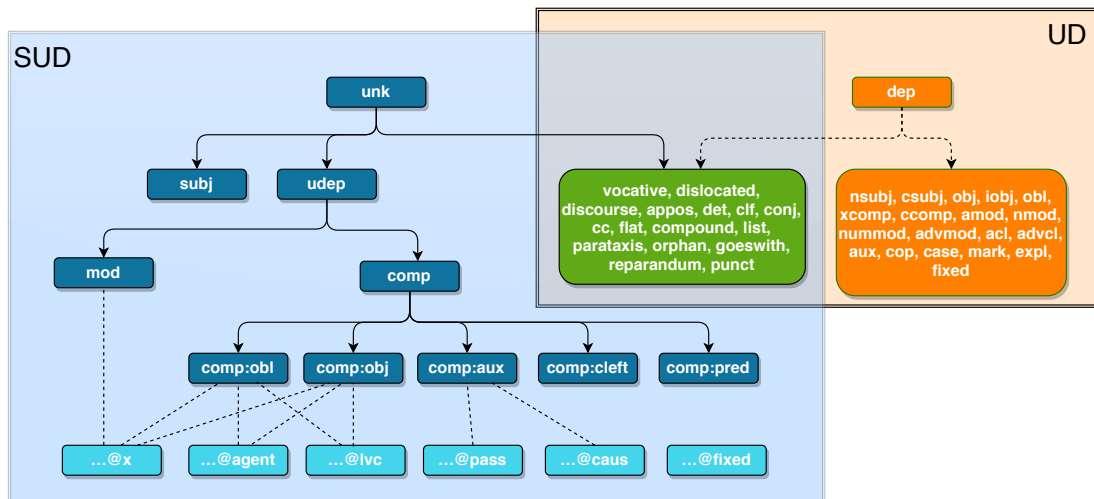


Figure 1: Taxonomy of SUD relations



Figure 2: Cleft sentences

3.1 Surface-syntactic relations

SUD has a unique subject relation, `subj`, and a unique relation `mod` for all modifiers. We will focus here on subrelations of the relation `comp`, for (subcategorized) complements.

- `comp:obj` is used for direct object complements (see examples in the previous section), including direct complements of an adposition or a subordinating conjunction: *about* >`comp:obj` *her*, *whether* >`comp:obj` *(she) leaves*.
- `comp:obl` is used for oblique complements, including clausal complements commuting with an adpositional complement (*I am afraid* >`comp:obl` *of your departure/to leave/that you leave*).
- `comp:pred` is used for predicative complements: *she is* >`comp:pred` *happy*; *she seems* >`comp:pred` *happy*; *I consider (her)* >`comp:pred` *happy*.
- `comp:aux` is used for the complement of an auxiliary: *she is* >`comp:aux` *sleeping*; *she has* >`comp:aux` *left*; (Fr) *elle fait* >`comp:aux` *dormir les enfants* [‘she makes the kids sleep’].
- `comp:cleft` is used for cleft clauses. In Figure 2, the first sentence resembles a relative clause more closely whereas the second sentence is impossible as a relative. Yet, both `comp:cleft` relations depend on *is*.

Due to the functional definition of SUD relations, the span of some UD relations is extended in SUD:

- `det` can be used with numerals in SUD, while all numerals must be `nummod` in UD. To retain the reversibility UD-SUD in the case of a numeral that functions as a determiner, we add a new UD subrelation `nummod:det`.
- Similarly, `discourse` can be used with verbs in SUD, while `parataxis` must be used in UD. In this case, the SUD `discourse` with a verbal dependent becomes `parataxis:discourse` in UD.

3.2 Deep-syntactic features

As explained in Section 2.2, we may have a predicate which does not have its own subject at the surface syntax level. The link of such a predicate with its semantic subject does not concern the surface

syntax but the syntax-semantics interface or what (Mel’čuk, 1988) calls the deep syntax. We decide to explicitly indicate this deep nature by introducing deep-syntactic features on dependencies with the @ symbol. In the two sentences *He wants to run* and *He came without running*, we introduce a feature @x: *wants* >comp:obj@x *to* >comp:obj *run*, *came* >mod@x *without* >comp:obj *running*.⁴ In other words, comp:obj@x is a comp:obj surface-syntactic relation whose verbal dependent has its deep subject somewhere in the sentence.⁵ This feature, which indicates that the dependent of the relation is not linked to its deep subject, is automatically subsumed by comp:pred and comp:aux and can be left out for these relations.

This strict separation between surface-syntactic relations and deep-syntactic features is extended to the conversion of other UD relations. For instance, a redistribution (diathesis change) can be signaled as follows:

- @pass indicates a passive construction (*she is* >comp:aux@pass *fascinated by his attitude*). It can also be borne by the subj relation when there is no auxiliary (for example *This business failed miserably, with many of the books* <subj@pass *sold as waste paper*.⁶).
- @caus indicates a causative construction: (Fr) *il fait* >comp:aux@caus *pleurer les enfants* [‘He made the kids cry’].
- @agent is used for a demoted subject: *she is fascinated* >comp:obl@agent *by his attitude*; (Fr) *il fait pleurer* >comp:obj@agent (*les*) *enfants* [‘he makes the kids cry’].

UD marks expletive elements with a dedicated relation expl. We consider that this is not a surface-syntactic relation, but it is possible to keep this information in the dedicated deep-syntactic feature @expl. See an example of an expletive subject in Figure 3.

Note that our annotation scheme remains centered around a surface syntactic analysis, but we isolate semantically-oriented features more explicitly. This allows for an easier interface with the Enhanced UD annotation effort (Schuster and Manning, 2016).



Figure 3: SUD analysis for *It is unlikely that she comes now*

Another example of deep-syntactic features is given by the annotation of light verb constructions: We use the @lvc deep-syntactic feature. It is a feature indicating that the dependent is a predicative noun and that the governor is a light verb without semantic contribution. Nouns in light verb constructions can have a comp:obl@x dependent.

- (Fr) *Avoir envie de manger* [‘having the urge to eat’]: *avoir* >comp:obj@lvc *envie* >comp:obl@x *de* >comp:obj *manger*;
- (Fr) *Avoir l’air heureuse* [‘having a happy appearance’]: *avoir* >comp:obj@lvc *air* >comp:pred@x *heureuse*;
- (Fr) *Mettre au défi de partir* [‘take on the challenge to leave’]: *mettre* >comp:obl@lvc *à* >comp:obj *défi* >comp:obl@x *de* >comp:obj *partir*.

4 Multi-words expressions in SUD

According to UD guidelines, a special relation fixed must be used to annotate some MWE: fixed grammaticized expressions that behave like function words or short adverbials. This notion of fixed expressions tries to take into account two aspects: the fact that there is no clear internal syntactic structure and

⁴We choose @x in reference to xcomp. We could use @y in case of the raising of an object as in *a book easy* >mod@y *to read*.

⁵Deep subjects are the first semantic argument of a verb or an adjective. They are labeled as subject in the enhanced UD graph, which is similar to the Mel’čukian deep-syntactic structure.

⁶https://en.m.wikipedia.org/wiki/Honoré_de_Balzac

the fact that the whole expression may have a POS which is not predictable from the POS of the internal tokens (Kahane et al., 2018). We would like to argue that these two aspects are not necessarily linked. In the sentence *He bought heaven knows what*, the idiomatic part *heaven knows what* has at the same time a clear internal syntactic structure and an unexpected POS in the context. SUD recommends an internal analysis of MWEs as soon as there are regular syntactic relations.

To take this into account, we propose to explicitly annotate the POS of a given expression when it is different from the POS of the head token. We propose in SUD to introduce the feature ExtPOS (for external POS) to give the POS of the whole expression.

In parallel, we also want to clearly indicated the span of the MWE; this must be done in the deep-syntactic layer because we can have a regular syntactic structure. In such cases, the span of the MWE is indicated by the deep-syntactic feature @fixed, added to the relation name (see Figure 4).

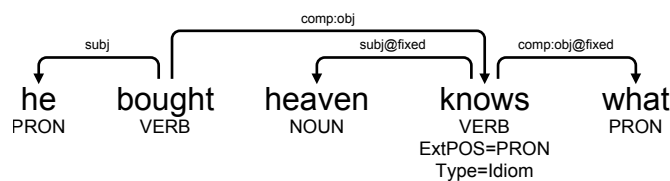


Figure 4: SUD analysis of an idiomatic construction

An alternative to this encoding is the token-based feature method applied in the PARSEME project (Savary and al., 2018; Kahane et al., 2018).

For phrases with no clear internal structure, we indicate at the surface-syntactic level (unk) the fact that the relation is unknown and at the deep-syntactic level (@fixed) that there is a fixed expression: *each >unk@fixed other, ad >unk@fixed hoc*.

It is interesting to observe that the fact that some phrase does not behave according to the POS of its head exists also in other contexts not related to MWEs. We also recommend the use of the ExtPOS feature in these cases, together with a Type feature to explicit the construction:

- In titles (of books, movies, songs...), the head can have various POS but it is most of the times used as a proper noun: *the movie Gone with the wind*, ExtPOS=PROPN, Type=Title.
- In grafts (Deulofeu, 1999; Deulofeu et al., 2010), which is a phenomenon mainly observed in spoken production, where a clause is used instead of a noun phrase: *he bought I think it is called dowels*, ExtPOS=NOUN, Type=Graft.

We also suggest that UD should adopt the ExpPOS feature or an equivalent mechanism. It will allow for easier generalizations and for more precise validation of the UD treebanks. For instance, in the current validation script of UD, the dependent advmod must be ADV unless it is a MWE, which means in UD, that the dependent has a fixed relation with one of its dependent. If UD adopted a feature-based encoding of MWEs, this condition could be replaced by the presence of ExtPOS=ADV, as in SUD.

5 UD ⇔ SUD conversion

The conversion SUD ⇒ UD is done in three main steps (Gerdes et al., 2018): 1) transforming the string analysis for the relation conj into a bouquet structure; 2) reversing relations comp:aux, comp:pred with an AUX governor, and comp:obj with an ADP, a SCONJ, or a PART governor (which gives us aux, cop, mark and case); 3) mapping other SUD relations directly to UD relations.

Two types of extensions in relations are considered:

- Some extensions are associated to special rules. For instance, comp:pred gives us xcomp (when the governor is not an AUX), as well as comp:obj@x and comp:obl@x.
- Deep-syntactic extensions are just copied as simple extensions, because the notion of deep-syntactic extension does not exist in UD. For instance, comp:aux@caus, gives us aux:caus.

In the UD \Rightarrow SUD conversion, the three same steps are applied.⁷ Extensions used in UD which are unknown in SUD are just copied but with the symbol @. For instance, `case:loc` used in different Chinese UD gives us `comp:obj@loc` in Chinese SUD.⁸

In order to avoid confusion between SUD and UD, relations which are common to the two annotation schemes have the same interpretation in both schemes. In some marginal cases (det or discourse), we allow a wider use for a relation in SUD.

It must be noted that there is not a one-to-one correspondence between SUD and UD relations, because the relations are defined on different principles. Nevertheless, in most cases, the conversion is reversible. For instance, UD `xcomp` corresponds to `comp:pred`, `comp:obj@x` and `comp:obl@x`, but in general the relation can be recovered according to the dependent's POS (ADJ or NOUN for `comp:pred`, VERB with or without a marker in the other cases).⁹ Conversely, the SUD `mod` relation corresponds to several UD relations according to the POS of the governor and the dependent (`amod`, `nummod`, `nmod`, `acl`, when the governor is a NOUN; `advmod`, `advcl`, `obl:mod` when the governor is a VERB). In case the `ExtPOS` feature is instantiated, it must be used for the determination of the UD relation, and not the regular POS feature.

6 Conclusion

The SUD principles have been further refined in this article:

- SUD must be translatable in UD, but SUD can be more precise than UD (cf. the case of UD `xcomp`).
- SUD tries to make a clear distinction between surface-syntax properties, only based on distributional criteria, and deep-syntactic properties, concerning the syntax-semantics interface.
- SUD needs an encoding of the POS of MWEs, since this is no longer encoded in the relation name.

SUD is available for the development of new treebanks. A github project dedicated to SUD is under construction at <https://surfacesyntacticud.github.io/>, which will collect all available resources for SUD: universal and language-specific annotation guidelines, natively annotated SUD treebanks, SUD treebanks automatically converted from UD, GREW grammars (Bonfante et al., 2018) for the conversion UD \Rightarrow SUD and SUD \Rightarrow UD, and other consolidation tools.

We hope that this alternative annotation scheme opens up the world of UD to communities that have been reluctant to adopt some UD annotation choices. Moreover, SUD is not only a well-grounded and validated annotation scheme that has been successively applied to languages of various language groups, the conversion tools and practice that we propose are designed for an easy deployment to other alternative annotation schemes around the UD project.

References

- Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*, volume 1 of *Logic, Linguistics and Computer Science Set*. ISTE Wiley.
- Bernard Caron, Marine Courtin, Kim Gerdes, and Sylvain Kahane. 2019. A surface-syntactic ud treebank for naija. In *Proceedings of Universal Dependencies Workshop*, Paris.
- Marine Courtin, Bernard Caron, Kim Gerdes, and Sylvain Kahane. 2018. Establishing a language by annotating a corpus: The case of naija, a post-creole spoken in nigeria. In *Proceedings of the workshop on Annotation in Digital Humanities (An-nDH)*, pages 7–11, Sofia.

⁷In this case, the second step is more delicate, because some elements must be raised to the functional head. For instance, in English, the negation is clearly borne by the auxiliary (*she has not slept*) because an auxiliary must always be present in case of negation (*she does not sleep*).

⁸In the conversion evaluated by (Gerdes et al., 2018), relations with unknown extensions were not treated, which was the source of many problems of non-reversibility.

⁹Yet, in some specific cases it is not possible to recover the SUD relation corresponding to the UD relation. For instance the `xcomp` relation in French can correspond to two different SUD relations for similar surface forms: *il rêve de venir* [‘he dreams of coming’] commutes with *il rêve de ça* [‘he dreams of that’] and is a `comp:obl@x`, while *il tente de venir* [‘he tries to come’] commutes with *il tente ça* [‘he tries that’] and is a `comp:obj@x`.

- Henri-José Deulofeu, Lucie Dufort, Kim Gerdes, Sylvain Kahane, and Paola Pietrandrea. 2010. Depends on what the french say: Spoken corpus annotation with and beyond syntactic function. In *Proceedings of the 4th Linguistic Annotation Workshop (LAW IV)*, Uppsala, Sweden.
- Henri-José Deulofeu. 1999. *Recherches sur les formes de la prédication dans les énoncés assertifs en français contemporain (le cas des énoncés introduits par le morphème que)*. Thèse d'état, Université Paris 3.
- Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.
- Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2018. SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In *Universal Dependencies Workshop 2018*, Brussels, Belgium, November.
- Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.
- Richard A. Hudson. 1984. *Word grammar*. Oxford: Blackwell.
- Richard A. Hudson. 1987. Zwicky on heads. *Journal of linguistics*, 23(1):109–132.
- Sylvain Kahane, Marine Courtin, and Kim Gerdes. 2018. Multi-word annotation in syntactic treebanks: Propositions for universal dependencies. In *Proceedings of the 16th international conference on Treebanks and Linguistic Theories*, Prague.
- Haitao Liu. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*.
- Igor Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. Albany, N.Y.: The SUNY Press.
- Joakim Nivre and 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.
- 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, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.
- Agata Savary and al. 2018. *PARSEME multilingual corpus of verbal multiword expressions*. Language Science Press.
- Sebastian Schuster and Christopher D. Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *LREC*.

Artificially Evolved Chunks for Morphosyntactic Analysis

Mark Anderson David Vilares Carlos Gómez-Rodríguez

Universidade da Coruña, CITIC

FASTPARSE Lab, LyS Research Group, Departamento de Computación

Campus Elviña, s/n, 15071

A Coruña, Spain

{m.anderson, david.vilares, carlos.gomez}@udc.es

Abstract

We introduce a language-agnostic evolutionary technique for automatically extracting chunks from dependency treebanks. We evaluate these chunks on a number of morphosyntactic tasks, namely POS¹ tagging, morphological feature tagging, and dependency parsing. We test the utility of these chunks in a host of different ways. We first learn chunking as one task in a shared multi-task framework together with POS and morphological feature tagging. The predictions from this network are then used as input to augment sequence-labelling dependency parsing. Finally, we investigate the impact chunks have on dependency parsing in a multi-task framework. Our results from these analyses show that these chunks improve performance at different levels of syntactic abstraction on English UD treebanks and a small, diverse subset of non-English UD treebanks.

1 Introduction

Shallow parsing, or chunking, consists of identifying constituent phrases (Abney, 1997). As such, it is fundamentally associated with constituency parsing, as it can be used as a first step for finding a full constituency tree (Ciravegna and Lavelli, 1999; Tsuruoka and Tsujii, 2005). However, chunking information can also be beneficial for dependency parsing (Attardi and DellOrletta, 2008; Tammewar et al., 2015), and vice versa (Kutlu and Cicekli, 2016). Latterly, Lacroix (2018) explored the efficacy of noun phrase (NP) chunking with respect to universal dependency (UD) parsing and POS tagging for English treebanks. As UD treebanks do not contain chunking annotation, they deduced chunks by adopting linguistic-based phrase rules. They observed improvements on POS and morphological feature tagging in a shared multi-task framework for the English treebanks in UD version 2.1 (Nivre et al., 2017). However, an increase in performance for parsing was only obtained for one treebank.

Contribution 1. We first relax the standard definition of chunks and present an evolutionary method to automatically deduce chunks for any language given a dependency treebank. 2. We show that chunking information can improve performances for POS tagging, morphological feature tagging, and dependency parsing, both in a multi-task and a single-task framework.

2 Chunks and chunking rules

While Lacroix (2018) described a method to obtain chunks from sentences with UD annotations, their approach is limited to NP chunks and requires hand-crafted linguistic rules, meaning that it cannot be transferred to other languages without language-specific knowledge. In contrast, we introduce a fully automatic approach to obtain chunks from UD-annotated sentences in a language-agnostic way. Figure 1 depicts our method of extracting candidate chunk types.

Chunk definition Here we loosen the definition of a chunk and consider any base-level subtree a possible chunk defined by the following criteria: (i) the components of a chunk are syntactically linked; (ii) there is only one level of dependency (one head and its dependents); (iii) the components are continuous; and (iv) no dependents within a chunk has a dependent outside the chunk.

¹POS tagging is used throughout to refer to universal part-of-speech (UPOS) tagging.

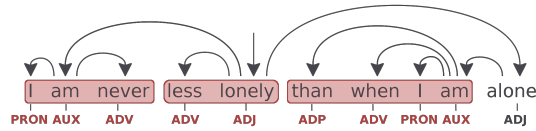


Figure 1: Candidate phrase rules are extracted by selecting subtrees with one level of dependency.

Describing chunks with rules For each subtree in the training set that meets the above criteria, the corresponding sequence of POS tags of its words is saved as a candidate rule. Each rule is collected for a given treebank to construct a ruleset of unique candidate chunk types. When more than one overlapping subtree meets these conditions the maximal substring is used, e.g. in Figure 1 PRON AUX ADV is chosen instead of PRON AUX or AUX ADV. We allow any chunk type with the exception of those containing the PUNC POS tag and we apply a mild frequency cut of 5 to make the problem more tractable. The English-EWT treebank, for example, results in a ruleset consisting of 512 candidates.

Annotating with rulesets This ruleset (or any subset of it) can be applied to a UD treebank to obtain chunks, by using them as patterns that generate a chunk when they are matched by a sequence of POS tags and meet the criteria described above.² In particular, we can apply it to the training set to obtain a set of chunks on which to train a statistical chunker to process arbitrary texts and help morphosyntactic tasks. When annotating a treebank, the POS tag of the head is used as a suffix for the chunk type, e.g. DET ADJ NOUN would result in IOB tags of B-NOUN and I-NOUN, assuming the head of this phrase corresponds to the NOUN tag (Ramshaw and Marcus, 1999).

However, not all candidate rules are useful and can impact the ability of a chunker to make sensible predictions. For this reason, we will not use the whole candidate ruleset obtained from a training corpus, but instead try to find a subset of the ruleset whose resulting set of chunks strikes a good balance between the following criteria: (i) coverage (i.e. there should be enough chunks to maximize their informativeness for morphosyntactic tasks) and (ii) consistency and learnability (i.e. the chunks should follow patterns predictable enough to be easily learnable by a machine learning model, so that our approach is not undermined by low chunking accuracy). Our hypothesis is that these two characteristics (which we quantify with a fitness function in the next section) are reasonable proxies for the usefulness of a particular set of chunks for morphosyntactic tasks.

Note that to achieve this, it is not possible to merely remove error-prone rules from the ruleset because there is a complicated interplay between rules, i.e. if the 10% most error-prone rules are removed, the overall accuracy of the system is not guaranteed to improve. Furthermore, with so many candidate rules, it is not possible to try every combination as this results in an astronomical number (2^n). Therefore, we aim to use an evolutionary method to find optimal subsets of rules to be used when annotating treebanks.

3 Evolutionary search for chunk rules

Evolutionary algorithms aim to optimise an objective (fitness) function by evaluating a population of individuals and subsequently generating a new population based on the best performing individuals from the population (Back, 1996). This process is then repeated until a set number of generations is reached or until the fitness function converges. Each individual consists of a set of parameters and its corresponding objective function value, or fitness. The fitness of an individual is used to decide whether to use it as a parent for subsequent generations or to remove it from the population. We introduce the techniques used to select parents and how they are then used to generate offspring (Algorithm 1 in Appendix A).

K-best parent selection The selection operator makes the population converge. We used the simple k-best method where the top k individuals of a population are selected as the parents.

Mutation Mutation is a genetic operator which prevents a population becoming too genetically similar by randomly altering individuals. This ensures that at least some level of genetic diversity is maintained

²Rules are applied from longer (more specific) to shorter (more generic).

from generation to generation. Our individuals have binary genes, so our mutation operator flips each gene with a probability $P_{\text{mutate gene}}$.

Crossover Crossover is a genetic operator which also preserves genetic variety in a population. In single-point crossover, a random index κ is chosen and the substring $0-\kappa$ of parent_x is replaced with the corresponding part of parent_y, and vice-versa. This results in two offspring. Single-point crossover can be extended to x-point crossover, where x points are used to cut individuals.

We used the DEAP framework for our implementation (Fortin et al., 2012), and the parameters in Table 6 (Appendix B). We represented our rulesets as a binary vector, where 1 meant a rule was used and 0 meant it was not. Our fitness function was obtained by combining the F1-score of a chunker implemented with the sequence-labelling framework NCRF++ (Yang and Zhang, 2018) and the proportion of the maximum compression rate, weighted 1.0 and 0.5 respectively. The compression rate, r , is defined as:

$$r = \frac{C_{\text{tokens}}}{C_{\text{chunks}} + C_{\text{out}}} \quad (1)$$

where C_{tokens} is the number of tokens in a treebank, C_{chunks} the number of chunks a ruleset creates, and C_{out} the number of tokens outside of chunks. And subsequently the proportion of the maximum compression rate, $r\%$ is defined as:

$$r\% = \frac{r_{\text{subset}} - 1}{r_{\text{all}} - 1} \quad (2)$$

where r_{subset} is the compression rate of the current rule subset and r_{all} is the compression rate of the full ruleset.

We used a small network for chunking due to the considerable computational costs of evolutionary algorithms. For each individual in each population, we trained a chunker for 5 epochs (see Table 7 in Appendix B for the parameters) and the corresponding model’s best performance on the development set was taken as that individual’s fitness along with the proportion of the maximum compression rate, $r\%$: the proportion of the maximum rate was used to prevent the algorithm from generating rulesets that generated few chunks and therefore minimising the potential impact. The convergence over 40 generations for English-EWT and Japanese-GSD can be seen in Figure 2.

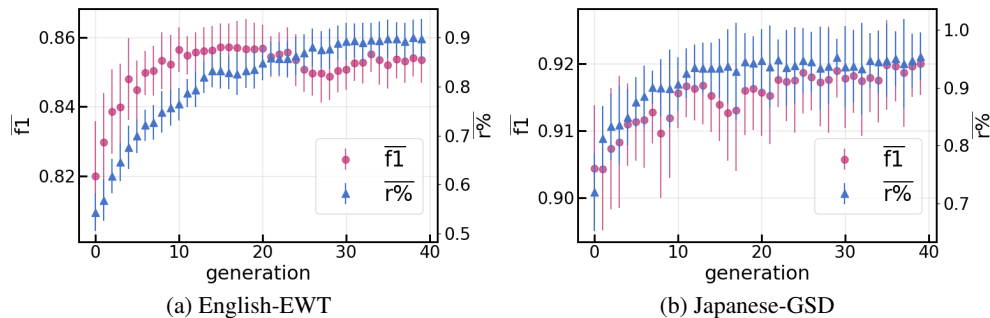


Figure 2: Average F1-score and proportion of max compression for English-EWT (a) and Japanese-GSD (b) during evolutionary search for optimal chunk type candidates.

As a final step, we took the top 100 best rulesets from across generations and extracted the rules that appeared in at least 75% and 95% of these sets, as the evolutionary algorithm only managed to find a single set with a fairly low performance. Rulesets were obtained this way for each treebank, except the rulesets extracted from English-EWT were subsequently used on the other English UD treebanks. The statistics for the resulting chunks for the respective test data can be seen in Table 1.

4 Sequence-labelling framework

All the proposed tasks can be cast as sequence labelling, so in this work we have used a sequence-labelling framework to address them. In particular, we rely on bidirectional long short-term memory

	# rules		C/sent	
	75%	95%	75%	95%
en-ewt	230	134	3.11	2.71
en-gum	-	-	4.48	3.84
en-lines	-	-	4.47	4.18
en-partut	-	-	6.32	5.84
bg	152	108	3.94	3.65
de	135	106	4.05	3.90
ja	184	130	6.83	6.70

Table 1: Chunking statistics on test data for each treebank used where # rules is the number of rules in a ruleset for a given threshold and C/sent corresponds to the number of chunks per sentence found.

(BiLSTMs) networks (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997). The input to the network are continuous word representations and character embeddings.

In this paper we used NCRF++ (Yang and Zhang, 2018), which uses stacked BiLSTMs, to generate contextualised hidden representations for every word (\vec{h}_i) in the input sentence. For decoding, it uses a feed-forward layer followed by a *softmax* activation:

$$P(y|\vec{h}_i) = \text{softmax}(\vec{W} \times \vec{h}_i + \vec{b}) \quad (3)$$

The single task models are optimised with cross-entropy loss, \mathcal{L} , defined as:

$$\mathcal{L} = - \sum \log(P(y|h_i)) \quad (4)$$

For the multi-task learning models, we implemented a hard-sharing architecture, where all the stacked BiLSTMs are shared across all tasks (Søgaard and Goldberg, 2016). A separate feed-forward layer (as the one used in the single task setup) is used to decode the output for each task. With respect to the computation of the loss under the multi-task learning (MTL) setup, \mathcal{L}_{MTL} , is defined as:

$$\mathcal{L}_{MTL} = \sum_{t \in T} \beta_t \mathcal{L}_t \quad (5)$$

where t is a task from the set of all tasks, T ; β_t is the corresponding weight for task t ; and \mathcal{L}_t is the cross-entropy loss for task t . A schematic of the network can be seen in Figure 3.

4.1 Dependency parsing as sequence labelling

In order to more readily utilise the multi-task framework for dependency parsing, we have cast dependency parsing as a sequence-labelling task. This was done by using the relative position encoding scheme introduced by Strzyz et al. (2019). We opted to use this encoding as it was the highest performing labelling scheme they evaluated. For each word in a sentence the dependency relation label is combined with the relative position of its head based on the POS tag of the head, e.g. a noun which is the subject of a verb (*son* in the input sentence in Figure 3) would have a label of +1,nsubj,VERB, where +1 indicates the head is the next VERB in the sentence and nsubj is the relation label.

5 Experiments

Data The analyses were undertaken using the English treebanks (EWT, GUM, LinES, and ParTUT) and also Bulgarian-BTB, German-GSD, and Japanese-GSD from UD v2.3 (Nivre et al., 2018). No results are given for Japanese-GSD for morphological feature tagging as it does not contain this information.

Network hyperparameters We used the framework as described above and hyperparameters from Vilares et al. (2019) which can be seen in Table 8 in the Appendix B. The standard input to the system consisted of word embeddings concatenated with character embeddings. All embeddings were randomly initialised.

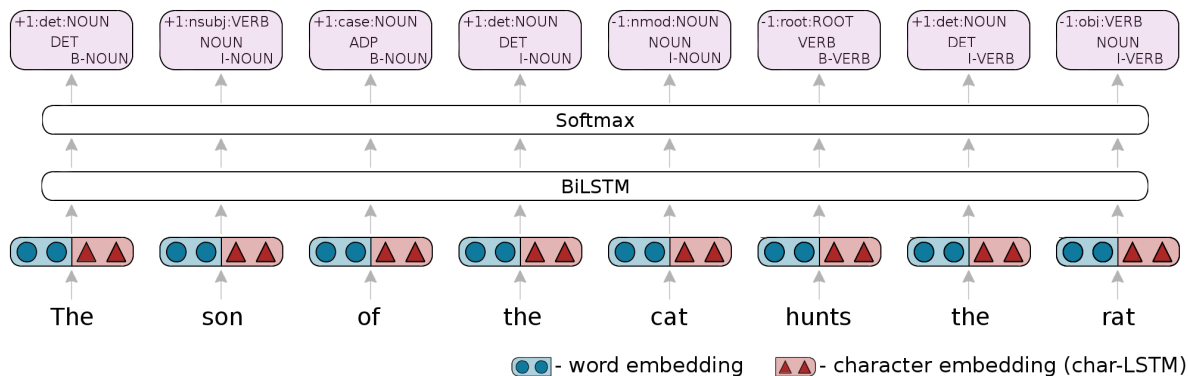


Figure 3: Multi-task architecture shown with sequence-labelling dependency parsing (as described in subsection 4.1), POS tagging, and chunking as shared tasks. Network input is a concatenation of word embeddings (circles) and character-level word embeddings (triangles) obtained from a character-based LSTM layer. The network is constructed of BiLSTM layers followed by a *softmax* layer for inference.

Experiment 1 We tested the impact of our chunks on POS and morphological feature tagging in a shared multi-task setting. This entails feeding word and character embeddings as input to the network with the output being some combination of POS tags, morphological feature tags, and chunk labels. These results were compared against the baseline taggers (single-task networks and POS and morphological features shared only). Tasks were equally weighted. As a further baseline we include results for POS and morphological feature tagging using UDPipe 2.2 (Straka and Straková, 2019).

Experiment 2 We used the best predictions (when using chunking) from experiment 1 as additional features for a sequence-labelling dependency parser (Strzyz et al., 2019). Therefore, network input consisted of word and character embedding and then some combination of POS tags, morphological feature tags, or chunk labels with the sole output being a dependency parser tag. We used gold tags and labels as input during training, but at runtime we used predicted tags and labels. For baselines we train a model with no features which is decoded with predicted POS tags using UDPipe 2.2 (as the sequence-labelling encoding we are using requires POS tags to resolve dependency heads) and also a model trained with POS tags as features but also using UDPipe 2.2 predicted POS tags at runtime.

Experiment 3 We tested the impact of our chunks on a sequence-labelling dependency parser in a multi-task framework with and without the other tasks. POS tagging was treated as a secondary main task with a weight of 0.5 (as POS tags are needed to decode the sequence-labelling scheme for the dependency parser) and chunks and morphological features were considered auxiliary tasks with a weight of 0.25 when used. The input during this experiment were only word and character embeddings. An example is shown in Figure 3 where the shared tasks are chunking, POS tagging, and dependency parsing. The baseline used here is a model trained solely to predict dependency parsing tags which are then decoded using predicted POS tags from UDPipe 2.2.

6 Results and discussion

As seen in Table 2 the multi-task framework with chunks improves the performance of both POS and morphological tagging for all English treebanks. In the same table, it is clear that they do not aid Bulgarian, but they do improve POS tagging performance for German and Japanese. Table 3 shows that chunking performance consistently improves in the multi-task setting. Parsing performance is improved across all treebanks when the predictions from experiment 1 are used as features (Table 4), but only for English-EWT (the largest treebank) and ParTUT (the smallest) do the predicted chunks explicitly improve performance and for the other treebanks only the other predicted features help. This is in contrast to the findings of Nguyen and Verspoor (2018), who obtained higher performance for larger treebanks. In the multi-task setting for the dependency parser (Table 5), the chunking information consistently aids

	ewt		gum		lines		partut	
	pos	feats	pos	feats	pos	feats	pos	feats
udpipe	94.44	95.37	93.88	94.21	94.73	94.83	94.10	94.01
single	95.08	96.09	94.61	94.92	95.64	95.57	94.69	94.54
pos+feats	95.23	96.21	94.60	95.26	95.59	95.71	94.63	94.16
pos+feats+chunks ₇₅	95.89	96.72	95.58	96.31	96.38	96.45	96.04	95.51
pos+feats+chunks ₉₅	95.86	96.52	95.52	96.21	96.35	96.33	96.21	95.60

	bg		de		ja	
	pos	feats	pos	feats	pos	feats
udpipe	97.78	95.55	92.03	70.18	96.39	-
single	97.41	95.06	93.07	87.14	96.97	-
pos+feats	97.69	94.84	92.90	87.28	-	-
pos+feats+chunks ₇₅	97.49	94.58	93.34	87.03	96.98	-
pos+feats+chunks ₉₅	97.44	94.45	92.90	87.11	97.09	-

Table 2: Multi-task tagging performance on English UD treebanks (en-ewt, en-gum, en-lines, and en-partut), Bulgarian-BTB (bg), German-GSD (de), and Japanese-GSD (ja) UD treebanks: single, single-task training; pos, with POS tagging; feats, with morphological feature tagging (except Japanese (ja) which has no morphological features); and chunks_x, with chunks with threshold x .

	baseline		multi	
	75%	95%	75%	95%
en-ewt	89.99	91.59	91.84	92.98
en-gum	85.76	88.11	88.08	89.98
en-lines	86.01	88.38	88.45	90.67
en-partut	88.36	90.78	91.79	93.30
bg	92.27	92.60	93.79	94.45
de	88.74	88.97	89.35	89.62
ja	93.35	92.73	94.39	94.02

Table 3: Chunker F1 scores in multi task setting where the baseline presented is from training the chunker for a given ruleset with threshold 75% or 95% as a single task and multi is from training with pos and morphological feature tagging except for Japanese (ja) which has no morphological features.

	en-ewt		en-gum		en-lines		en-partut	
	uas	las	uas	las	uas	las	uas	las
no features ^{udpipe}	80.97	77.87	76.70	72.71	76.43	71.87	81.63	78.67
pos ^{udpipe}	84.88	81.79	81.09	76.87	79.06	74.08	84.01	80.63
pos	86.15	83.29	83.03	79.31	80.76	76.12	85.83	82.69
pos-feats	86.32	83.37	82.83	79.13	81.15	76.48	86.71	83.60
pos-chunks ₇₅	85.84	82.87	82.49	78.83	80.86	76.04	87.03	83.86
pos-chunks ₉₅	85.80	82.86	81.95	78.19	80.32	75.55	86.65	83.36
pos-feats-chunks ₇₅	86.43	83.41	82.61	78.86	81.13	76.21	87.09	83.86
pos-feats-chunks ₉₅	85.99	83.04	82.15	78.50	80.82	76.09	87.35	84.04

	bg		de		ja	
	uas	las	uas	las	uas	las
no features ^{udpipe}	86.49	82.43	63.20	58.86	89.96	88.43
pos ^{udpipe}	89.48	85.30	79.39	74.04	92.49	90.42
pos	89.47	85.11	81.77	76.69	93.68	91.70
pos-feats	89.74	85.48	82.05	77.12	-	-
pos-chunks ₇₅	89.23	84.67	81.49	76.54	93.28	91.41
pos-chunks ₉₅	89.06	84.77	81.55	76.40	92.95	91.20
pos-feats-chunks ₇₅	89.11	84.83	81.77	76.71	-	-
pos-feats-chunks ₉₅	89.24	85.07	81.41	76.38	-	-

Table 4: Feature input ablation for dependency parser with English UD treebanks (en-ewt, en-gum, en-lines, and en-partut), Bulgarian-BTB (bg), German-GSD (de), and Japanese-GSD (ja) UD treebanks: no features^{udpipe}, no features but UDPipe predicted POS tags used to decode; pos, gold POS tags for training and predicted POS tags for runtime (pos^{udpipe} UDPipe predicted POS tags used); feats, gold morphological feature tags for training and predicted feature tags for runtime; and chunks_x, gold chunks with threshold x at training time and predicted chunks for runtime.

	en-ewt		en-gum		en-lines		en-partut	
	uas	las	uas	las	uas	las	uas	las
single ^{udpipe}	80.97	77.87	76.70	72.71	76.43	71.87	81.63	78.67
pos	84.52	81.30	78.94	74.96	78.75	74.13	83.66	80.25
pos-feats	84.21	81.14	79.51	75.42	78.56	73.87	84.10	81.31
pos-chunks ₇₅	84.55	81.51	79.54	75.48	78.17	73.55	83.86	81.13
pos-chunks ₉₅	84.42	81.34	79.60	75.54	78.72	74.20	83.57	80.16
pos-feats-chunks ₇₅	84.25	81.24	79.81	75.84	78.75	73.95	84.01	80.90
pos-feats-chunks ₉₅	84.24	81.18	79.48	75.36	78.84	74.15	84.98	81.92

	bg		de		ja	
	uas	las	uas	las	uas	las
single ^{udpipe}	86.49	82.43	63.20	58.86	89.96	88.43
pos	88.00	83.89	80.75	75.59	93.25	91.45
pos-feats	88.07	83.89	80.46	75.50	-	-
pos-chunks ₇₅	87.90	83.66	81.29	75.96	93.25	91.61
pos-chunks ₉₅	88.07	83.93	80.98	75.71	93.04	91.28
pos-feats-chunks ₇₅	88.26	84.00	80.77	75.52	-	-
pos-feats-chunks ₉₅	88.09	83.67	80.69	75.63	-	-

Table 5: Multi-task parsing results for English (en-ewt, en-gum, en-lines, and en-partut), Bulgarian-BTB (bg), German-GSD (de), and Japanese-GSD (ja) UD treebanks: single^{udpipe}, parsing as single task with UDPipe predicted POS tags used to decode parser output; pos, with POS tagging as aux. task; feats, with morphological feature tagging as aux. task; and chunks_x, with chunking as aux. task for threshold x .

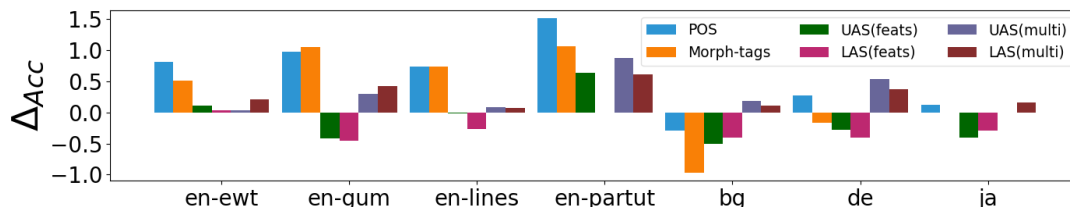


Figure 4: Difference in accuracy for each task between the best model with chunks and the best without.

performance with a meaningful increase in accuracy observed over baseline models for each treebank.

As can be seen in Figure 4, the change in performance when using the predicted chunks as a feature for parsing is less profound than in the multi-task experiments. Only two English treebanks explicitly benefit from predicted chunks, whereas all treebanks benefit from at least one feature. So the performance is at least implicitly improved by using our chunks, except for the more morphologically-rich (especially with respect to verbal inflection) Bulgarian. The treebank used for Japanese, generally an agglutinative language, does not contain morphological features, so perhaps it too would not improve with chunks if they could have been used. Therefore, it would be interesting to evaluate whether the impact of chunking information is predicated by certain linguistic features. Furthermore, the increase in performance for each treebank for the multi-task experiments suggests that the performance when using the chunks as input would improve with better predicted chunks, which corroborates the findings of Lacroix (2018).

7 Conclusion

We have introduced a language-agnostic method for extracting chunks from dependency treebanks. We have also shown the efficacy of these chunks with respect to improving POS tagging, morphological feature tagging, and dependency parsing for a number of UD treebanks.

Acknowledgments

This work has received funding from the European Research Council (ERC), under the European Unions Horizon 2020 research and innovation programme (FASTPARSE, grant agreement No 714150), from the ANSWER-ASAP project (TIN2017-85160-C2-1-R) from MINECO, and from Xunta de Galicia (ED431B 2017/01). We thank one anonymous reviewer for in-depth comments and suggestions.

References

- S. Abney, 1997. *Part-of-Speech Tagging and Partial Parsing*, pages 118–136. Springer Netherlands, Dordrecht.
- Gluseppe Attardi and Felice Dell’Orletta. 2008. Chunking and dependency parsing. In *Proceedings of LREC Workshop on Partial Parsing: Between Chunking and Deep Parsing*.
- Thomas Back. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.
- Fabio Ciravegna and Alberto Lavelli. 1999. Full text parsing using cascades of rules: an information extraction perspective. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*.
- Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research*, 13:2171–2175, jul.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Mucahit Kutlu and Ilyas Cicekli. 2016. Noun phrase chunking for turkish using a dependency parser. In *Information Sciences and Systems 2015*, pages 381–391. Springer.
- Ophélie Lacroix. 2018. Investigating NP-Chunking with Universal Dependencies for English. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 85–90.
- Dat Quoc Nguyen and Karin Verspoor. 2018. An improved neural network model for joint POS tagging and dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 81–91, Brussels, Belgium, October. Association for Computational Linguistics.
- Joakim Nivre, Željko Agić, Lars Ahrenberg, et al. 2017. Universal Dependencies 2.1. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, et al. 2018. Universal Dependencies 2.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.
- Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.
- Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany, August. Association for Computational Linguistics.
- Milan Straka and Jana Straková. 2019. Universal dependencies 2.4 models for UDPipe (2019-05-31). LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. Viable dependency parsing as sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 717–723, Minneapolis, Minnesota, June. Association for Computational Linguistics.
- Aniruddha Tammewar, Karan Singla, Bhasha Agrawal, Riyaz Bhat, and Dipti Misra Sharma. 2015. Can distributed word embeddings be an alternative to costly linguistic features: A study on parsing hindi. In *Proceedings of the 6th Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2015)*, pages 21–30.
- Yoshimasa Tsuruoka and Jun’ichi Tsujii. 2005. Chunk parsing revisited. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 133–140, Vancouver, British Columbia, October. Association for Computational Linguistics.

David Vilares, Mostafa Abdou, and Anders Søgaard. 2019. Better, faster, stronger sequence tagging constituent parsers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3372–3383, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Jie Yang and Yue Zhang. 2018. NCRF++: An Open-source Neural Sequence Labeling Toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Appendix A Evolutionary algorithm

Algorithm 1 Evolutionary algorithm

```

1: for gen  $\leftarrow$  maxgen do
2:   for ind in population do
3:     ind.fit  $\leftarrow$  GETFITNESS(ind)
4:   end for
5:   offspring  $\leftarrow$  SELECT(population)
6:   offspring  $\leftarrow$  CLONE(offspring)
7:   for pair in offspring2i, offspring2i+1 do
8:     if random  $<$  Pcrossover then
9:       pair  $\leftarrow$  CROSSOVER(pair)
10:    end if
11:  end for
12:  for ind in offspring do
13:    if random  $<$  Pmutate then
14:      ind  $\leftarrow$  MUTATE(ind)
15:    end if
16:  end for
17:  population  $\leftarrow$  offspring
18: end for

19: function GETFITNESS(ind)
20:   rules  $\leftarrow$  CONVERT(ind)
21:   train, dev  $\leftarrow$  CHUNKTREEBANKS(rules)
22:   TRAINCHUNKER(train)
23:   F1  $\leftarrow$  EVALULATECHUNKER(dev)
24:   Rp  $\leftarrow$  GETMAXRPROPORTION(dev)
25:   return F1 + 0.5·Rp
26: end function

```

Appendix B Hyperparameters

hyperparameter	value
population size	100
number of generations	4
k-best	5
P _{mutate}	0.5
P _{mutate gene}	0.05
P _{crossover}	0.5
decay (linear)	0.1

Table 6: Hyperparameters for the evolutionary algorithm: k-best, the number of best parents chosen to seed next generation; P_{mutate}, the probability an individual will mutate; P_{mutate gene}, the probability a given gene will mutate; P_{crossover}, the probability a pair of individuals will crossover; and decay is how much P_{mutate} and P_{crossover} decrease after each generation.

hyperparameter	value
BiLSTM dimensions	200
BiLSTM layers	1
word embedding dimensions	50
character embedding dimensions	30
character hidden dimensions	50
character CNN layers	4
CNN window size	3
optimiser	SGD
loss function	cross entropy
learning rate	0.015
decay (linear)	0.05
momentum	0.9
dropout	0.5
L ₂ regularisation	1×10^{-8}
epochs	5
training batch size	10
runtime batch size	128

Table 7: Hyperparameters for the neural-net chunker used during the evolutionary algorithm.

hyperparameter	value
BiLSTM dimensions	800
BiLSTM layers	2
word embedding dimensions	100
character embedding dimensions	30
character hidden dimensions	50
feature dimensions	20
optimiser	SGD
loss function	cross entropy
learning rate	0.2
decay (linear)	0.05
momentum	0.9
dropout	0.5
epochs	100
training batch size	8
runtime batch size	128

Table 8: Hyperparameters for the network used in all experiments.

Challenges of language change and variation: towards an extended treebank of Medieval French

Mathilde Regnault^{1,2} and Sophie Prévost¹ and Eric Villemonte de la Clergerie²

mathilde.regnault@sorbonne-nouvelle.fr

sophie.prevost@ens.fr

Eric.De_La_Clergerie@inria.fr

(1) Lattice, 1 rue Maurice Arnoux, 92120 Montrouge, France

(2) Inria, 2 rue Simone Iff, 75012 Paris, France

Abstract

In order to automatically extend a treebank of Old French (9th-13th c.) with new texts in Old and Middle French (14th-15th c.), we need to adapt tools for syntactic annotation. However, these stages of French are subjected to great variation, and parsing historical texts remains an issue. We chose to adapt a symbolic system, the French Metagrammar (FRMG), and develop a lexicon comparable to the Lefff lexicon for Old and Middle French. The final goal of our project is to model the evolution of language through the whole period of Medieval French (9th-15th c.).

1 Introduction

With the rise of digital humanities, more and more ancient texts are made available. Annotating them and keeping this information in treebanks helps study and describe old stages of a language. Some are available for Medieval French (9th-15th c.), namely the MCVF¹ (Martineau, 2008), annotated with constituency syntax, and the SRCMF² (Prévost and Stein, 2013), annotated with dependency syntax and covering Old French (9th-13th c.) for now. Our goal is to automatically extend the SRCMF treebank to obtain a larger resource. In particular, we want to add texts of Middle French, the next stage in the evolution of French (14th-15th c.), as well as new texts of Old French. This new resource would then contain one million words, four times more than the current SRCMF. We want to annotate these data automatically with the highest quality, which means we need to find a way to parse both Old and Middle French. However, this task is difficult because we have limited resources annotated with dependency syntax in Old French, and none in Middle French (Guibon et al., 2014). Moreover, Medieval French, like Old French, is subjected to great variation (sections 2 and 3).

The new texts will be annotated by both a statistical and a symbolic parser. The annotation will then be merged to obtain the best possible analysis. For this work, we focus on the symbolic approach. Using wide coverage grammars (Oepen et al., 2004; Rocio et al., 2003; Brants et al., 2002) has shown effective (section 4), so we chose to adapt the French Metagrammar (FRMG, Villemonte de la Clergerie (2005; 2013)), a symbolic system for contemporary French (section 5). Our contribution is to make a diachronic grammar for Medieval French, currently still in process.

2 Parsing historical texts

The extension of the SRCMF treebank will contain not only new texts in Middle French, but also in Old French, which will give a more accurate view of the period. Six dialects from the northern half of the territory will be added, and the representation of the different domains and genres will be better balanced owing to the new texts. For example, there is only one historical work in the latest version of the SRCMF, which prevents from drawing thorough comparisons with other domains.

However, the heterogeneity of these data is challenging. An automatic system is very unlikely to give the correct analysis of phenomena that had few occurrences (or none) in the training data. A grammar is

¹The MCVF treebank (*Modéliser le changement : les voies du français*) is available at this address: <http://www.voies.uottawa.ca/>.

²The SRCMF treebank (Syntactic Reference Corpus of Medieval French) is available at this address: <http://srcmf.org/>.

also subjected to such limitation because developers rely on existing descriptions and treebanks. Moreover, the amount of available data annotated with dependency syntax is limited, and only exists for Old French. We considered several methods to solve this parsing challenge. Guibon et al. (2014) investigated statistical parsing of Old French on the SRCMF with the *Mate* parser (Bohnet, 2010). They obtained an average labelled attachment score (LAS) of 76.04. They developed a methodology relying on the selection of a training set with metadata³ similar to the new text to annotate with a minimum error rate (Guibon et al., 2015).

Although it would be possible to use such a method to extend the SRCMF treebank, statistical parsing still heavily depends on the training data. The solution we chose is to adapt a symbolic or hybrid system built for French, following Rocio et al. (2003) for Old Portuguese. However, that system will have to be flexible enough to enable the treatment of the great variability of Medieval French.

3 Difficulties due to the specificities of Medieval French

Even though Contemporary French largely differs from Medieval French, there are still enough similarities to enable us to adapt a grammar. Almost all syntactic phenomena in French are already present in Medieval French, but with different frequencies. For example, SVO became the prevalent word-order as early as the 11th century (it was previously SOV, inherited from Latin, which was prevalent). In case of complete absence of information on words, it is possible to resort to a morphological and syntactic lexicon of contemporary French. The descriptions of syntactic phenomena should however be modified to parse Medieval French and cope with linguistic variation.

From a synchronic perspective, Medieval French is characterised by a great variability. First of all, it has a free word-order and null subjects. Latin had a nominal declension to help determine the syntactic function of words, but it started to decline very early in Medieval French, and soon became inefficient, as well as rich verbal endings. Buridant (2010) gives this example from *Lancelot, the Knight of the Cart*:

- (1) Lancelot Vit la dame de la maison
Lancelot saw the lady of the house

where both *Lancelot* and *la dame de la maison* are candidates for the subject position. The context helps determine their roles: it is the lady who is subject of the sentence. But the grammar we are developing does not have the capability of deducing dependencies from the context.

Furthermore, many dialects are included, which have an impact on the frequencies of occurrences of word forms and syntactic phenomena. The spelling of words was not fixed, even in a same dialect, which leads to the presence of different writings of the same words in a single text. This makes their recognition and analysis more complex. Finally, due to the different forms and domains of texts and the individual styles of authors, different frequencies of phenomena and words can be observed. This should also be taken into account while choosing the datasets to train a disambiguation model. Unlike contemporary languages, synchronic variation in historical texts can be difficult to define because there is no "standard language" we can describe first and extend with the specificities which are encountered in other texts. The number of resources is also limited, which may cause biases in an analysis.

From a diachronic perspective, texts are also subjected to variation. Frequencies of the different word-orders and constructions have evolved through time. These evolutions are however not linear. For example, the OSV order was very rare in the 13th century, and it peaked in the 14th and 15th centuries (Marchello-Nizia, 2008; Combettes and Prévost, 2015).

The valency of some words (i.e. the number and types of argument they require), especially verbs, has evolved too. For example, *morir* (to die) could be a transitive, meaning in this case "to kill", but it is strictly intransitive in contemporary French. Evolution of word sense and use can be observed within Medieval French and has an impact on syntactic analysis because their distributions are different at each period.

Variation is a salient property of Medieval French. It appears at many levels and needs to be handled by parsers.

³Some characteristics, like dialect, are more discriminative than others.

4 Related work

Several treebanks for ancient languages are available, for example Latin (Bamman and Crane, 2006), Old English (Taylor, 2007), Medieval Portuguese (Rocio et al., 2003) and Middle High German (Hinrichs and Zastrow, 2012). Other annotated corpora can be found in the CLARIN Research Infrastructure (Hinrichs and Krauwer, 2014). They can be diachronic, which makes the annotation challenging because of morphological and syntactic changes.

The MCVF treebank is the biggest treebank for Medieval French, with 361.283 words for Old French alone, against 251.000 in the SRCMF treebank. Although adjustments must be done in order to adapt the annotation scheme to ours, its size and the presence of texts of Middle French make it a promising resource for machine learning techniques, some of which seem more appropriate for this kind of data.

For example, transfer learning is nowadays used for low-resource languages (Agić et al., 2016). Provided that we develop a parallel corpus for Medieval French, this technique can be explored. It is still possible to do cross-language transfer without such parallel data, as Scrivner and Kübler (2012) did for Old Occitan, a language from the South of France and close to Old French. They chose modern Catalan as their source language for syntax because the word-order is "relatively free", as in Old Occitan. We can use a treebank of Contemporary French, but it is likely to introduce a bias in favour of an analysis close to the modern language. We would not be able to constrain the syntactic models according to linguistic knowledge.

We can also consider using automatic normalisation as an additional layer of annotation, because it has shown efficient for historical texts (Bollmann and Sjøgaard, 2016). This too is to be explored in a machine learning approach.

This work focuses on a symbolic approach, which will be compared to statistical parsing later on. Brants et al. (2002) pointed out an advantage of parsing with a grammar: the annotation is consistent and has high accuracy. Some projects were successful in adapting existing systems to former stages of a language, as discussed earlier. The extension of the LinGO Redwoods treebank should also be mentioned (Oepen et al., 2004; Toutanova et al., 2005). The authors use a HPSG grammar for analysis and statistical models for disambiguation, ensuring the coherence of annotation. Our grammar should also enable us to annotate new texts following the existing treebank's scheme.

5 Solutions for syntactic analysis

In order to parse Medieval French, we chose to adapt FRMG because of the modularity and flexibility a metagrammar provides.

5.1 French Metagrammar

A metagrammar (Candito, 1996) consists of a hierarchy of small classes describing the rules underlying a grammar. It is a mean to factorise linguistic description, therefore making maintenance and corrections easier. A first general description of a phenomenon is written in a "mother class", from which more specific classes inherit. The metagrammar is compiled into a grammar, which is then used by a parser.

FRMG is a metagrammar based on the Tree Adjoining Grammar (TAG) formalism (Joshi et al., 1975), extended with feature structures. These grammars use elementary trees as units, which have a finite depth and are associated with an item of the lexicon. They are combined to build whole sentences using a non-contextual operation, substitution, and a contextual one, adjunction. A TAG is mildly context-sensitive. In FRMG's implementation of TAGs (Villemonte de la Clergerie, 2010), some operators have been added, like disjunction, Kleene star (mainly for coordinations), or guards, expressing conditions on nodes. Sibling nodes are not ordered, which is useful for the analysis of a language with a free word-order.

FRMG's feature structures are hypertags (Kinyon, 2000), a unique structure containing the information of the elementary trees a word can anchor. This is equivalent to a set of supertags. They include grammatical category, sub-categorisation and semantic type. The Lefff lexicon has compatible hypertags with FRMG, which enables the metagrammar to request the information needed for the syntactic analysis, such as POS-tag, gender, number, valency, and the possible forms of the expected arguments of

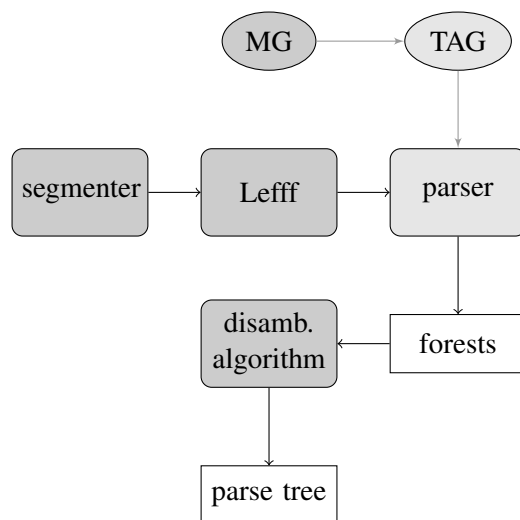


Figure 1: Architecture of the pipeline developed by Boullier et al. (2005)

words (verbs, nouns, adjectives...). If an elementary tree is incompatible with the processed sentence, it is discarded. Having all information available for each word does not cause too much ambiguity.

After the segmentation (see Fig. 1, Boullier et al. (2005)), word lattices are enriched with information from the morphological and syntactic lexicon. This enables to preserve ambiguities. The parser produces all possible analysis in the form of a shared forest of derivation trees, then converted into dependency trees. The disambiguation model selects the most probable solution. It is trained on the French Treebank (FTB) (Abeillé et al., 2003) for contemporary French (Villemonte de la Clergerie, 2013). We will use the SRCMF for our system. Following Guibon et al. (2015), training data should be split according to the metadata of texts, so that the weights in the model fit the new texts to parse. We also consider automatically classifying texts, which would help making use of texts with uncertain metadata or no assigned dialect.

5.2 Adapting FRMG

We use the pipeline described above to parse Medieval French. OFrLex, a lexicon similar to the Lefff (Sagot, 2010), is under development (Sagot, 2019). It includes a new kind of information to add to entries: spelling variants. All variants of a word are linked to it, which is useful for a language with no strict notion of "orthography".

The adaptation of FRMG to Medieval French is a work in progress divided into four main steps. We chose to develop only one metagrammar for the whole period because it is not possible for us to accurately describe each state of language separately. There is no clear boundary between them, they tend to overlap. We consider at first that their main difference is the distribution of frequencies of words and syntactic constructions. As language evolution is not linear, some declining phenomena may rise again some decades later, preventing a straight-forward modelling of language change. Medieval French may be considered as a succession of states of language preparing the contemporary French, but with much more variation and some looser rules, as it can be observed for verbal agreement. Some nouns can either be considered as singular or plural because of their nature as "collective", like *gent* (people).

ex. from *Alexis*: *crient la gent*, transl. "people scream"

(2) crient la gent
VERB (pl) DET (sg) NOUN (sg)

ex. from *Roland*: *La gent de France iert blecee e blesmie*, transl. "The people of France were hurt and turned pale"

(3) La gent de France iert blecee ...
 DET (sg) NOUN (sg) ADP NOUN **VERB (sg)** VERB (sg) ...

Our first step towards the adaptation of FRMG is therefore to loosen these constraints, at least for

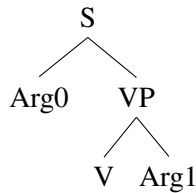


Figure 2: Organisation of the main constituents in FRMG

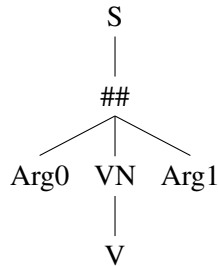


Figure 3: Organisation of the main constituents in our metagrammar. The node ## permits a free order between its children nodes.

collective nouns. Since the states of language are subjected to great variation, all possible analysis should be enabled by the metagrammar until we find out new specific constraints. Missing descriptions also need to be added. We want to be as close to FRMG as possible, keeping most of the descriptions and all the types, to build a continuum between the states of language.

Secondly, in order to deal with free word-order, we chose to change the description of the main constituents. FRMG has a traditional tree representation of a canonical sentence (see Fig. 2), while we chose a flatter representation (see Fig. 3), as advocated by Abeillé et al. (2003). The extension of the TAG formalism permits free order between sibling nodes, which makes our descriptions simpler. Otherwise, we would have to create multiple attachments for verbal arguments in the sentence tree. For example, we find SVO order in *Yvain*, as analysed in the SRCMF:

- (4) messire Gauvains ainme Yvain
 my lord Gauvain likes Yvain

In the same text, we also find VSO order, which is analysed with the same elementary tree (see Fig. 4):

- (5) ainme ele li
 likes she him

Thirdly, we want to develop a new mechanism to handle language variation. After all possibilities of analysis are described, we want to restrain some constructions according to the metadata of texts, like the dialect, the genre or the period. The date of a text is particularly informative about the syntax. Some

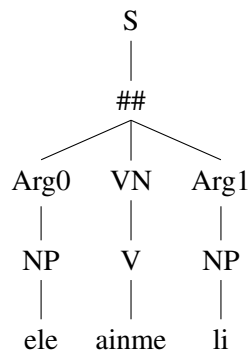


Figure 4: Analysis of sentence (5)

syntactic constructions are known to appear or disappear at a certain period. For example, the OSV order was possible only with a subject pronoun until the 13th century (Schøsler, 1984). By including such a constraint in the metagrammar, we reduce the ambiguity on many sentences. Specificities of dialects have also been described in previous work. For instance, object clitics are usually found before the verb. Some are however found after, but only in texts written in *picard*, a dialect from the North, as in this example from *Escouffe*, v. 4954-55, as cited by Buridant (2010):

(6) prestés me huimais L'ostel
offer me for today hospitality

These special rules can be found in traditional grammars, but we plan to search for new ones with error mining (Sagot and Villemonte de la Clergerie, 2006). Adding a facet handling these exceptions will enable us to describe a general case, instead of under-specifying descriptions in order to enable all possible realisations.

6 Perspectives

We want to extend the SRCMF with the highest quality. Annotation should remain coherent with its annotation scheme. For this purpose, we are currently adapting a large coverage grammar to Medieval French. It has to be completed to be evaluated on a whole corpus and not only on single sentences. This system will then be compared to statistic and neural approaches. This work also aims at developing a methodology for the analysis of heterogeneous data in general, such as tweets and forums.

Acknowledgements

This work takes place in the ANR-16-CE38-0010 PROFITEROLE project (2017–2020), directed by Sophie Prévost.

References

- Anne Abeillé, Lionel Clément, François Toussnel. 2003. Building a treebank for French. In *Treebanks: Building and Using Parsed Corpora*. Kluwer Academic Publishers.
- Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter and Anders Søgaard. 2016. Multilingual Projection for Parsing Truly Low-Resource Languages. *Transactions of the Association for Computational Linguistics*.
- David Bamman and Gregory Crane. 2006. The design and use of a Latin dependency treebank. *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT2006)*, p. 67–78.
- Bernd Bohnet. 2010. Very high accuracy and fast dependency parsing is not a contradiction. *The 23rd International Conference on Computational Linguistics (COLING 2010)*. Beijing, China.
- Marcel Bollmann and Anders Søgaard. 2016. Improving historical spelling normalization with bi-directional LSTMs and multi-task learning. In *COLING*.
- Pierre Boullier, Lionel Clément, Benoît Sagot and Eric Villemonte de la Clergerie. 2005. Chaînes de traitement syntaxique. In *TALN 05*, p. 103–112. Dourdan, France.
- Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. *Proceedings of the workshop on treebanks and linguistic theories*, vol. 168.
- Claude Buridant. 2010. *Nouvelle Grammaire de l'ancien français*. Paris: Sedes.
- Marie-Hélène Candito. 1996. A principle-based hierarchical representation of LTAGs. *Proceedings of COLING-96*. Copenhagen, Denmark.
- Bernard Combettes and Sophie Prévost. 2015. La disparition du schéma V2 en français : le rôle de l'opposition marqué / non marqué dans le domaine syntaxique. In *Disparitions. Contributions à l'étude du changement linguistique*, T. Verjans and C. Badiou-Monferran. p. 283–301. Paris: Champion.

- Gaël Guibon, Isabelle Tellier, Mathieu Constant, Sophie Prévost and Kim Gerdes. 2014. Parsing Poorly Standardized Language Dependency on Old French. *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)*, V. Henrich and E. Hinrichs and D.de Kok and P. Osenova and A. Przepiórkowski, p. 51–61. Tübingen, Germany.
- Gaël Guibon, Isabelle Tellier, Sophie Prévost, Mathieu Constant and Kim Gerdes. 2015. Searching for Discriminative Metadata of Heterogenous Corpora. *Proceedings of the Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT14)*.
- Erhard Hinrichs and Thomas Zastrow. 2012. Automatic Annotation and Manual Evaluation of the Diachronic German Corpus TüBa-D/DC. *LREC*, p.1622–27.
- Erhard Hinrichs and Steven Krauer. 2014. The CLARIN Research Infrastructure: Resources and Tools for E-Humanities Scholars. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, p.1525–31.
- Aravind K. Joshi, Leon S. Levy and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences*.
- Alexandra Kinyon. 2000. HYPERTAGS: Beyond POS Tagging. *Natural Language Processing NLP 2000*, D. N. Christodoulakis. Springer-Verlag Berlin Heidelberg.
- Christiane Marchello-Nizia. 2008. *L'évolution de l'ordre des mots en français : Chronologie, périodisation, et réorganisation du système*, in Congrès Mondial de Linguistique Française, Paris.
- France Martineau. 2008. Un corpus pour l'analyse de la variation et du changement linguistique. *Corpus*, 7, <http://journals.openedition.org/corpus/1508>.
- Stephan Oepen, Dan Flickinger, Kristina Toutanova and Christopher D. Manning. 2004. LinGO Redwoods: A Rich and Dynamic Treebank for HPSG. *Research on Language and Computation*, vol. 2, p. 575–596.
- Sophie Prévost and Achim Stein. 2013. *Syntactic Reference Corpus of Medieval French (SRCMF)*, version 0.92. <http://srcmf.org>. ENS de Lyon; Lattice, Paris; ILR University of Stuttgart.
- Vitor Rocio, Mário Amado Alves, J. Gabriel Lopes, Maria Francisca Xavier and Graça Vicente. 2003. Automated Creation of a Medieval Portuguese Partial Treebank. *Treebanks: Building and Using Parsed Corpora*, Anne Abeillé. Kluwer Academic Publishers.
- Benoît Sagot. 2010. The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French. *7th international conference on Language Resources and Evaluation (LREC 2010)*.
- Benoît Sagot. 2019. Développement d'un lexique morphologique et syntaxique de l'ancien français. *26ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN)*. Toulouse, France.
- Benoît Sagot and Eric Villemonte de la Clergerie. 2006. Error mining in parsing results. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL - ACL '06*.
- Lene Schøsler. 1984. *La déclinaison bicasuelle de l'ancien français : son rôle dans la syntaxe de la phrase, les causes de sa disparition*. Odense University Press.
- Olga Scrivner and Sandra Kübler. 2012. Building an old Occitan corpus via cross-Language transfer. *KONVENS*, p. 392–400.
- Ann Taylor. 2007. The YorkTorontoHelsinki Parsed Corpus of Old English Prose. *Creating and Digitizing Language Corpora: Volume 2: Diachronic Databases*.
- Kristina Toutanova, Christopher D. Manning, Dan Flickinger and Stephan Oepen. 2005. Stochastic HPSG Parse Disambiguation using the Redwoods Corpus. *Research on Language and Computation*, vol. 3, p. 83–105.
- Eric Villemonte de la Clergerie. 2005. From metagrammars to factorized TAG/TIG parsers. *Proceedings of the Ninth International Workshop on Parsing Technology - Parsing '05*.
- Eric Villemonte de la Clergerie. 2010. Building factorized TAGs with meta-grammars. *The 10th International Conference on Tree Adjoining Grammars and Related Formalisms - TAG+10*.
- Eric Villemonte de la Clergerie. 2013. Improving a symbolic parser through partially supervised learning. *The 13th International Conference on Parsing Technologies (IWPT)*. Naria, Japan.

