Depling 2021

Sixth International Conference on Dependency Linguistics (Depling, SyntaxFest 2021)

Proceedings

To be held as part of SyntaxFest 2021
21–25 March, 2022
Sofia, Bulgaria

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 978-1-955917-14-8

Preface

The Sixth edition of the International Conference on Dependency Linguistics (Depling 2021) follows a biannual series that started in 2011, in Barcelona, and continued in Prague (2013), Uppsala (2015), Pisa (2017), and Paris (2019). The series responds to the growing need for linguistic meetings dedicated to approaches in syntax, semantics, and the lexicon that are centered around dependency structures as a central linguistic notion. For the second time, Depling is part of SyntaxFest, which co-locates four related but independent events:

- The Sixth International Conference on Dependency Linguistics (Depling 2021)
- The Second Workshop on Quantitative Syntax (Quasy 2021)
- The 20th International Workshop on Treebanks and Linguistic Theories (TLT 2021)
- The Fifth Workshop on Universal Dependencies (UDW 2021)

The reasons that suggested bringing these four events together in 2019 still hold in 2021. There is a continuing, strong interest in corpora and dependency treebanks for empirically validating syntactic theories, studying syntax from quantitative and theoretical points of view, and for training machine learning models for natural language processing. Much of this research is increasingly multilingual and cross-lingual, made in no small part possible by the Universal Dependencies project, which continues to grow at currently nearly 200 treebanks in over 100 languages.

For these reasons and encouraged by the success of the first SyntaxFest, which was held in 2019 in Paris, we – the chairs of the four events – decided to bring them together again in 2021. Due to the vagaries of the COVID-19 pandemic, it was eventually decided to push the actual SyntaxFest 2021 back to March 2022. In order not to delay the publication of new research and not to conflict with other events, we decided however to publish the proceedings that you are now reading in advance, in December 2021.

As in 2019, we organized a single reviewing process for the whole SyntaxFest, with identical paper formats for all four events. Authors could indicate (multiple) venue preferences, but the assignment of papers to events for accepted papers was made by the program chairs.

38 long papers were submitted, 25 to Depling, 11 to Quasy, 17 to TLT, and 24 to UDW. The program chairs accepted 30 (79%) and assigned 8 to Depling, 5 to Quasy, 7 to TLT, and 10 to UDW. 22 short papers were submitted, 6 to Depling, 7 to Quasy, 9 to TLT, and 9 to UDW. The program chairs accepted 14 (64%) and assigned 3 to Depling, 3 to Quasy, 3 to TLT, and 5 to UDW.

At the time of this writing, we do not yet know whether SyntaxFest will be a hybrid or purely online event. We regret this uncertainty but are nevertheless looking forward to it very much. Our sincere thanks go to everyone who is making this event possible, including everybody who submitted their papers, and of course the reviewers for their time and their valuable comments and suggestions. We would like to thank Djamé Seddah, whose assistance and expertise in organizing SyntaxFests was invaluable. Finally, we would also like to thank ACL SIGPARSE for its endorsement and the ACL Anthology for publishing the proceedings.

Radek Čech, Xinying Chen, Daniel Dakota, Miryam de Lhoneux, Kilian Evang, Sandra Kübler, Nicolas Mazziotta, Simon Mille, Reut Tsarfaty (co-chairs)

Petya Osenova, Kiril Simov (local organizers and co-chairs)

December 2021

Program co-chairs

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

- Depling:
  - Nicolas Mazziotta (Université de Liège)
  - Simon Mille (Universitat Pompeu Fabra)
- Quasy:
  - Radek Čech (University of Ostrava)
  - Xinying Chen (Xi'an Jiaotong University)
- TLT:
  - Daniel Dakota (Indiana University)
  - Kilian Evang (Heinrich Heine University Düsseldorf)
  - Sandra Kübler (Indiana University)
- UDW:
  - Miryam de Lhoneux (Uppsala University / KU Leuven / University of Copenhagen)
  - Reut Tsarfaty (Bar-Ilan University / AI2)

Local Organizing Committee of the SyntaxFest

- Petya Osenova (Bulgarian Academy of Sciences)
- Kiril Simov (Bulgarian Academy of Sciences)

Program Committee for the Whole SyntaxFest

Patricia Amaral (Indiana University Bloomington)
Valerio Basile (University of Turin)
David Beck (University of Alberta)
Ann Bies (Linguistic Data Consortium, University of Pennsylvania)
Xavier Blanco (UAB)
Igor Boguslavsky (Universidad Politécnica de Madrid)
Bernd Bohnet (Google)
Cristina Bosco (University of Turin)
Gosse Bouma (Rijksuniversiteit Groningen)
Miriam Butt (Universität Konstanz)
Marie Candito (Universtité Paris 7 / INRIA)
Radek Cech (University of Ostrava)
Giuseppe Giovanni Antonio Celano (University of Pavia)
Xinying Chen (Xi'an Jiaotong University)
Silvie Cinková (Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics)
Cagri Coltekin (University of Tuebingen)
Benoit Crabbé (Université Paris 7 / Institut national de recherche en informatique et en automatique, Paris)
Daniel Dakota (Indiana University)
Eric De La Clergerie (Institut national de recherche en informatique et en automatique, Paris)
Felice Dell'Orletta (Institute for Computational Linguistics, National Research Council, Pisa)
Kaja Dobrovoljc (Jožef Stefan Institute)
Kilian Evang (Heinrich Heine University Düsseldorf)
Thiago Ferreira (University of São Paulo)
Ramon Ferrer-I-Cancho (Universitat Politècnica de Catalunya)
Kim Gerdes (Laboratoire Interdisciplinaire des Sciences du Numérique, Université Paris-Saclay)
Carlos Gómez-Rodríguez (Universidade da Coruña)
Jan Hajic (Institute of Formal and Applied Linguistics, Charles University, Prague)
Eva Hajicova (Institute of Formal and Applied Linguistics, Charles University, Prague)
Dag Haug (University of Oslo)
Richard Hudson (University College London)
András Imrényi (Eszterházy Károly Egyetem)
Leonid Iomdin (Institute for Information Transmission Problems, Russian Academy of Sciences)
Jingyang Jiang (Zhejiang University)
Sylvain Kahane (Modyco, Université Paris Ouest Nanterre / CNRS)
Vaclava Kettnerova (Institute of Formal and Applied Linguistics)
Sandra Kübler (Indiana University Bloomington)
Guy Lapalme (University of Montreal)
François Lareau (Observatoire de linguistique Sens-Texte, Université de Montréal)
Alessandro Lenci (University of Pisa)
Nicholas Lester (University of Zurich)
Lori Levin (Carnegie Mellon University)
Haitao Liu (Zhejiang University)

Marketa Lopatkova (Institute of Formal and Applied Linguistics, Charles University, Prague)
Olga Lyashevskaya (National Research University Higher School of Economics)
Teresa Lynn (Dublin City University)
Jan Macutek (Mathematical Institute of the Slovak Academy of Sciences / Constantine the Philosopher University in Nitra)
Robert Malouf (San Diego State University)
Alessandro Mazzei (Dipartimento di Informatica, Università di Torino)
Nicolas Mazziotta (Université de Liège)
Alexander Mehler (Text Technology Group, Goethe-University Frankfurt am Main)
Wolfgang Menzel (Department of Informatics, Hamburg University)
Jasmina Milicevic (Dalhousie University)
Simon Mille (Pompeu Fabra University)
Yusuke Miyao (The University of Tokyo)
Simonetta Montemagni (Institute for Computational Linguistics, National Research Council, Pisa)
Kaili Müürisep (University of Tartu)
Alexis Nasr (Laboratoire d'Informatique Fondamentale, Université de la Méditerranée, Aix-Marseille II)
Sven Naumann (University of Trier)
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 (Zhejiang University)
Petya Osenova (Sofia University / Institute of Information and Communication Technologies, Sofia)
Robert Östling (Department of Linguistics, Stockholm University)
Agnieszka Patejuk (Polish Academy of Sciences / University of Oxford)
Alain Polguère (Université de Lorraine)
Prokopis Prokopidis (Institute for Language and Speech Processing/Athena RC)
Laura Pérez Mayos (Pompeu Fabra University)
Owen Rambow (Stony Brook University)
Rudolf Rosa (Institute of Formal and Applied Linguistics, Charles University, Prague)
Tanja Samardzic (University of Zurich)
Giorgio Satta (University of Padua)
Nathan Schneider (Georgetown University)
Olga Scrivner (Indiana University Bloomington)
Djamé Seddah (Alpage, Université Paris la Sorbonne)
Alexander Shvets (Institute for Systems Analysis of Russian Academy of Sciences)
Maria Simi (Università di Pisa)
Achim Stein (University of Stuttgart)
Reut Tsarfaty (Faculty of Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot)
Francis M. Tyers (Indiana University Bloomington)
Zdenka Uresova (Institute of Formal and Applied Linguistics, Charles University, Prague)
Gertjan Van Noord (University of Groningen)
Giulia Venturi (Institute for Computational Linguistics, National Research Council, Pisa)
Veronika Vincze (Hungarian Academy of Sciences, Research Group on Articial Intelligence)
Relja Vulanovic (Kent State University at Stark)

Chunshan Xu (anhui jianzhu university)
Xiang Yu (University of Stuttgart)
Zdenek Zabokrtsky (Institute of Formal and Applied Linguistics, Charles University, Prague)
Amir Zeldes (Georgetown University)
Daniel Zeman (Institute of Formal and Applied Linguistics, Charles University, Prague)
Hongxin Zhang (Zhejiang University)
Yiyi Zhao (Institute of Applied Linguistics, Communication University of China, Beijing)
Heike Zinsmeister (University of Hamburg)
Miryam de Lhoneux (University of Copenhagen)
Marie-Catherine de Marneffe (The Ohio State University)
Valeria de Paiva (Samsung Research America and University of Birmingham)

Additional Reviewers

Chiara Alzetta (National Research Council of Italy)
Aditya Bhargava (University of Toronto)
Lauren Cassidy (Dublin City University)
Simon Petitjean (Heinrich Heine University Düsseldorf)
Xenia Petukhova (National Research University Higher School of Economics)
Daniel Swanson (Indiana University)
He Zhou (Indiana University)
Yulia Zinova (Heinrich Heine University Düsseldorf)

Table of Contents

A Dependency Treebank for Classical Arabic Poetry 1
Sharefah Al-Ghamdi, Hend Al-Khalifa and Abdulmalik Al-Salman

On auxiliary verb in Universal Dependencies: untangling the issue and proposing a systematized annotation strategy 10
Magali Duran, Adriana Pagano, Amanda Rassi and Thiago Pardo

How useful are Enhanced Universal Dependencies for semantic interpretation? 22
Jamie Y. Findlay and Dag T. T. Haug

Starting a new treebank? Go SUD! 35
Kim Gerdes, Bruno Guillaume, Sylvain Kahane and Guy Perrier

Mutual dependency and Word Grammar: headedness in the noun phrase 47
Nikolas Gisborne

A monarchy without subjects: on Brassai's (almost) subject-free dependency grammar 60
András Imrényi

BINGO: A Dependency Grammar Framework to Understand Hardware Specifications Written in English 68
Rahul Krishnamurthy and Michael S. Hsiao

Drawing the syntactic space: choices in diagrammatic reasoning 81
Nicolas Mazziotta

Causation (and Some Other) Paraphrasing Patterns in L1 English. A Case Study 91
Jasmina Milićević

Is one head enough? Mention heads in coreference annotations compared with UD-style heads . . . 101
Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský and Daniel Zeman

Number agreement, dependency length, and word order in Finnish traditional dialects 115
Kaius Sinnemäki and Akira Takaki

A Dependency Treebank for Classical Arabic Poetry

Sharefah Al-Ghamdi
King Saud University
sharefah@ksu.edu.sa

Hend Al-Khalifa
King Saud University
hendk@ksu.edu.sa

Abdulmalik Al-Salman
King Saud University
salman@ksu.edu.sa

Abstract

This paper introduces the first syntactically annotated corpus for Classical Arabic poetry, a morphologically rich ancient Arabic text. The paper describes how the dependency treebank was prepared, focusing on some issues dealing with Classical Arabic poems in which syntactic constructions require special attention. We also present the results of the baseline experiments on Classical Arabic poetry dependency parsing with this treebank.

1 Introduction

With the massive development of natural language processing (NLP) applications and tools, treebanks (TB) (syntactically parsed text corpora) are considered an essential basic language resource. The existence of a treebank is the first step toward parser creation and evaluation for any natural language. Unfortunately, classical Arabic (CA) has only one treebank, which is for the Holy Quran text (Dukes and Buckwalter, 2010). This motivated us to contribute to the Arabic NLP resources by constructing the first Arabic Poetry Treebank (ArPoT).

CA (aka Quranic Arabic) is the standardized literary form of the Arabic language; It consists of the Holy Quran text and literary texts such as poetry, elevated prose, and oratory. However, it differs in its vocabulary and phraseology from the Modern Standard Arabic (MSA) that came with the prevalence of literacy, universal education, journalism, and written media. Moreover, CA poems are characterized by symmetry, eloquence, and rhetoric (Zwettler, 1978; Ahmed and Trausan-Matu, 2017). To maintain the rhyme and rhythm of poems, poets would violate the grammatical requirements showing, called the Poetic Necessity (Najjar, 2012). Thus, this work explores the dependency syntactic analysis of CA poems, and we expect that it would be a starting point for further studies on CA poetry parsing.

For our annotation scheme, we have chosen the part of speech (POS) tag sets, dependency labels and guidelines released by Habash et al. (2009), which have been applied during constructing Columbia Arabic Treebank (CATiB). We selected this schema based on two considerations. First, it is closer to the traditional Arabic grammar; however, it maintains the ability to do a future conversion to other different representations such as Universal dependency (UD) (Habash et al., 2009; Taji et al., 2017). Second, there is a publicly available parser that trained on Columbia Arabic Treebank, which we used in the initial annotation step. So that it would simplify and speed up the development process.

This paper describes the annotation process and outlines some of the issues and interesting phenomena found during the annotation of ArPoT. The rest of the paper is structured as follows: Section 2 briefly reviews the Arabic treebanks. Section 3 introduces the dataset that has been used to construct the ArPoT. Next, the annotation process is described in Section 4. Then, Section 5 discusses the challenges and issues we had tackled. Finally, we present the results of the baseline parsing experiments on our treebank in Section 6, and conclude the paper with future work in Section 7.

2 Related Work

Most of the well-known syntactic Arabic TBs are constructed for MSA, such as: constituency Penn Arabic Treebank (PATB) by Maamouri et al. (2004), Prague Arabic Dependency Treebank (PADT) by Hajic et al. (2004) and dependency Columbia Arabic Treebank (CATiB) by Habash et al. (2009). For

CA, Quranic Arabic Dependency Treebank (QADT) of the Holy Quran text by Dukes et al. (2010) is the only known TB. Its linguistic framework is termed a hybrid dependency-phrase structure grammar and focuses more on visualizing the grammatical annotation. The syntactic layer of QADT covers 37,578 words (~ 49% of the full Quranic text) (Dukes and Habash, 2011).

In addition to the above, several TBs for Arabic dialects have been produced, such as: Levantine Arabic Treebank (LATB) (Maamouri et al., 2006), Egyptian Arabic Treebank (Maamouri et al., 2014), and dependency treebank of Arabic tweets (Albogamy et al., 2017). However, there is no Arabic poetry Treebank that has been created yet.

Example 1: يا عَينِ جُـودي بِالدموع   المُستَهِلّاتِ السـوَافِحْ

yaA ðayni juwudiy bi AlddmwuðI   Almusthil~aAti AsswaAifH.

"Oh eye, be generous with shedding and pouring tears"

Figure 1. Classical Arabic verse

3 Dataset Preparation

3.1 Poems collection

Poems in ArPoT have been collected initially from Arabic literary poems websites such as ADAB[1] and ALDIWAN[2]. They offer thousands of written poems for transmitted oral poetry from the earliest pre-Islamic era until today. For this work, we only focus on Classical poetry, which commonly refers to old oral poems transmitted from the early (6th to 13th) centuries. The selected verses are diverse; they are from more than 775 poems for 34 different Classical eras poets. Our final corpus contains 2685 verses (35,459 tokens).

Classical verses consist of two parts that follow the metric rule, which is not the case of modern free poetry verses. Figure 1 shows example 1 for Classical verse along with its transliteration[3] and English translation. In addition, Table 1 lists a word for word glosses for all examples of CA verses that used in this work.

Word	Gloss	Word	Gloss	Word	Gloss	Word	Gloss
Example 1		**Example 3**		**Example 4**		**Example 5**	
يا	oh	و	and	فإن	if	خلاءين	empty
عين	eye	رب	rare	أبك	I cry	بعد	after
جودي	be generous	شمعة	candle	قومي	my people	الحلم	meekness
ب	with	مزقت	tears up	يا	oh	و	and
الدموع	tears	ثوب	garment	نوار	proper name	الجهل	rudeness
المستهلات	shedding	الظلام	darkness	ف	so	في	in
السوافح	pouring	ب	with	إني	I	هما	them
Example 2		ما	that	أرى	see	و	and
فيضا	flood	بثت	spread	مسجدي	mosques	بعد	after
كما	as	من	of	هم	their	عبابي	roar
فاض	flood	النور	light	من	of	الندى	rain
الغروب	pails	في	in	هم	them	المتدافع	rush
المترعات	generous	الأرجاء	surroundings	ك	as	-	-
من	from	متسعا	widely	البلاقع	desolate home	-	-
النواضح	camels						

Table 1. A word for word glosses for all examples of CA verses.

[1] https://www.adab.com/

[2] https://www.aldiwan.net/

[3] All Arabic transpirations are according to (Habash et al., 2007) transliteration scheme.

3.2 Preprocessing

In this stage, we have prepared the poetry text for annotation. After verses had been scraped from the webpages into text files, we concatenated the two parts of each verse using our implemented java code. Then, the spelling mistakes were corrected manually. During this phase, we removed the identical verses which are accidentally repeated on the websites. Also, there were some verses that were clearly broken and had several missing words shown as dots. The syntactic structure analysis for such verses was not able, so we removed them from the dataset. The "التطويل/ *Atatweel*/ Kashedah" has been removed as well. Since the verses are from transmitted old oral classical poems, the punctuation is uncommon and very rare. Therefore, the punctuation has been eliminated in this dataset.

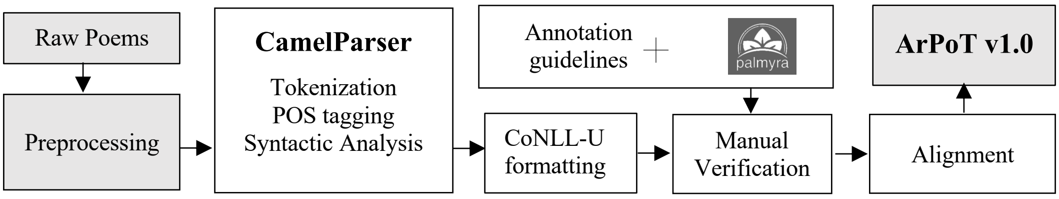


Figure 2. Annotation Process of ArPoT

4 Annotation process

To maintain the annotation process cost (in terms of money and time), we considered the strategy of automatic annotation followed by manual correction instead of creating the Arabic Poetry Treebank from scratch. Figure 2 shows the flowchart for the annotation steps.

4.1 Initial automatic annotation

After reviewing the dependency parsers for the Arabic language, we chose the CamelParser (Shahrour et al., 2016) for the initial automatic annotation. It is a publicly available system for Arabic syntactic dependency analysis that is trained on CATiB (Habash et al., 2009). Although it was developed on MSA, its initial parsing shortened the annotation process. It applies the tokenization and POS tagging with reasonable accuracy, and it constructs the syntactic trees we provide to the annotators for manual corrections.

4.2 File Format transformation

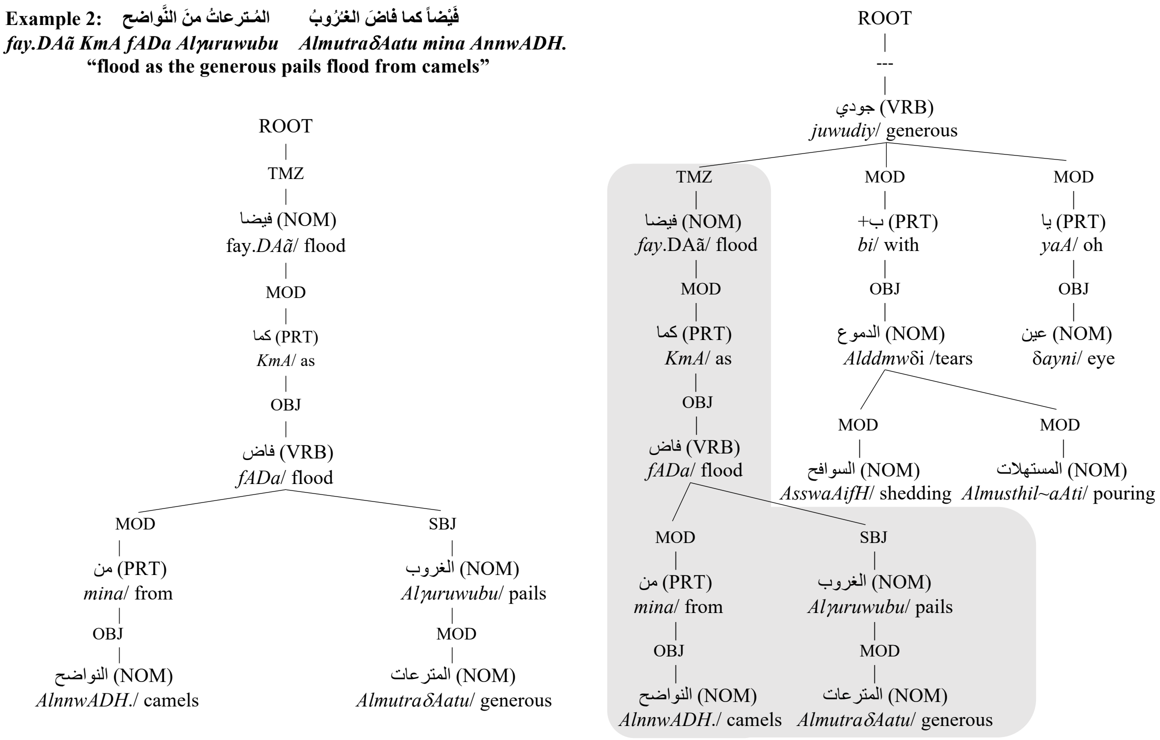
The CamelParser offers the output in different formats. However, we decided to produce a valid CoNLL-U format that can train most of the current parsers and tree visualization tools.

4.3 Manual Verification

While CamelParser was trained on MSA corpus, it handles the CA poems with tokenization, POS tagging, and dependency relation labeling errors. The manual correction phase starts with correcting the tokenization errors to give the ability to calculate the Inter Annotation Agreement (IAA) between the annotators. Three paid annotators have carried out this phase. They were Arabic native speakers and linguistic experts. PALMYRA, a graphical dependency tree visualization and editing software, has been used for this step (Javed et al., 2018; Taji and Habash, 2020). The manual correction was completed within four months.

CamelParser's tokenization was incorrect for around 52% of words. Thus, to report its accuracy on the CA poems, we compared the verses that have true tokenization with the final gold annotated verses which were verified by annotators. The result gave us 55% Exact Match (EM) – the percentage of tokens with correct POS tags, heads and relation labels.

We used the Kappa coefficient for IAA between annotators (Cohen, 1960). The first part of the data, which covers ~ 83% of the corpus, was revised by two full-time annotators with a 0.97 kappa value on 10% of this part. To check the agreement, the second part of the data, which covers ~17% of the corpus plus 10% of the first part, has been revised by a third annotator. The result of IAA was 0.85 for the



(a) Before Alignment

(b) After Alignment (Example 1 and 2)

Figure 3. The Alignment for two contiguous verses that have dependency relation in between.

kappa coefficient; then, after the second round of revision, the IAA increased to 0.96. The small size of the data and the few tags included in the guidelines positively affected the agreement score. Moreover, the CATiB annotator's manual provided to the annotators decreased the disagreement cases.

4.4 Alignment

Like the Quranic text, CA poetry consists of verses, which might be one complete sentence. However, the verse may act as a modifier for prior or posterior verse so that the complete sentence would be in two, three or more verses. Although sentence boundary detection is essential for NLP, there is no available system that could detect the sentence boundaries of the CA poetry. Therefore, we concatenated the verses' dependency trees for the same sentence during the alignment phase. Moreover, delaying the alignment step after the manual verification has simplified the visualization during the correction, while large trees after alignment become more complicated.

During the manual verification, we added a syntactic label to the root in case it has a relation with another verse and recorded the index of the parent token. Then, in the alignment phase, we just connected the related verses to produce one complete sentence in one syntactic tree. This broad tree shows the whole meaning that the verses will provide. For example, Figure 3 (a) shows the dependency tree for verse example 2 which is the subsequent of verse example 1 in the same poem (shown in Figure 1). The head token of example 2 syntactic tree has TMZ "تمييز/ tamyiz/ specification" relation with the word "جودي/ juwudy/ be generous" in the verse example 1. After the alignment for these contiguous verses to form a complete sentence, the connected tree for verse example 2 is shown with gray shade in Figure 3 (b).

5 CA Poetry Annotation Issues

Although the main guiding principle followed during the construction of ArPoT v1.0 serves as a general guideline, some syntactic structure issues and phenomena of CA poetry have been encountered. In the following, we present two categories of issues along with the solution strategies we applied.

Example 3: و رُبّ (*) شمعةٍ مزّقت ثوبَ الظلامِ بما   بثّت من النّورِ في الأرجاءِ متسعا

wa šamδahĩ maz~aqat θawba AĎĎalaAami bi maA   baθ~at mina Annwri fi AlÂrjaA'i mutasaδaA

“Rarely that a candle tears up the darkness garment, with its light that has been spread widely in the surroundings”

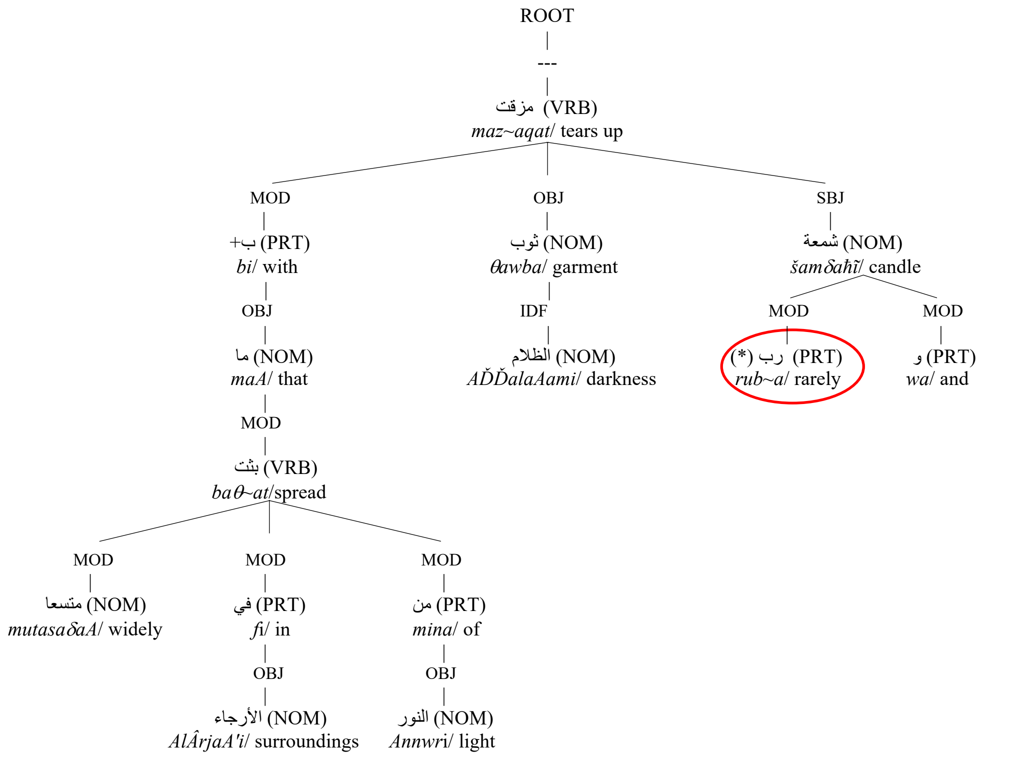


Figure 4. Dependency tree for the verse example 3 that shows the elision case.

5.1 Elision and Reconstruction

Linguistic deletion or elision (الحذف/ AlHaðf) is a common syntax feature in Classical Arabic language, mainly in Quranic text and poetry, where a major element of the sentence is omitted but often implied and recovered based on contextual clues (Suleiman, 1990). On the other hand, the process of allowing implicit syntactic roles to be made explicit is known as reconstructing (التقدير/ Altaqdir). Adding the ellipse to the sentence structure through reconstruction provides new information or meaning which unable to clarify except with (التقدير/ Altaqdir). Thus, we followed Dukes and Buckwalter (2010) in their treatment of elision cases by showing the empty nodes in the syntactic tree. In ArPoT, only 0.6% of the tokens are ellipses. During the manual verification, annotators added those dropped words manually to the treebank in the form (word (*)).

Ellipsis in ArPoT includes different categories such as: verbs, subject of nominal sentences, and particles deletion. For example, the deleted preposition (رُبَّ/ rub~a) has been added to the verse syntactic tree of verse example 3 as shown in Figure 4. In this example, (رُبَّ/ rub~a) gives the meaning of (التقليل/ taqliyl/ reduction), which means it is rare that one candle can give that much light.

The preposition (رُبَّ/ rub~a) is obviously used in CA. In the Arabic language, it is known as a semi-extra preposition (حرف شبيه بالزائد). This means that it illustrates the sentence's meaning, but it does not relate to its object like other original prepositions. Thus, we attached it under its object with MOD relation.

Example 4: فإنْ أبْكِ قَوْمي يا نَوَارُ فإنّني   أرَى مَسْجِدَيهِمْ مِنهمُ كالبَلاقعِ

faĂn Âbki qawmiy yaA naw~aAru faĂn~aniy   Âraý masjidayhim minhumu kaAlbalaAaqiδi

“If I cry my people, oh Nawwar, that because I see their mosques as desolate home”

Example 5: خَلاءَينِ بَعدَ الحِلْمِ وَالجَهلِ فيهما   وَ بَعْدَ عُبابيِّ النّدَى المُتَدافعِ

xalaA'ayni baδda AlHilmi wa Aljahli fihimaA   wa baδda δubabiy~ Annadý AlmutadaAfiδi

“I see them empty, after meekness and rudeness there, and after the roar of heavy rain”

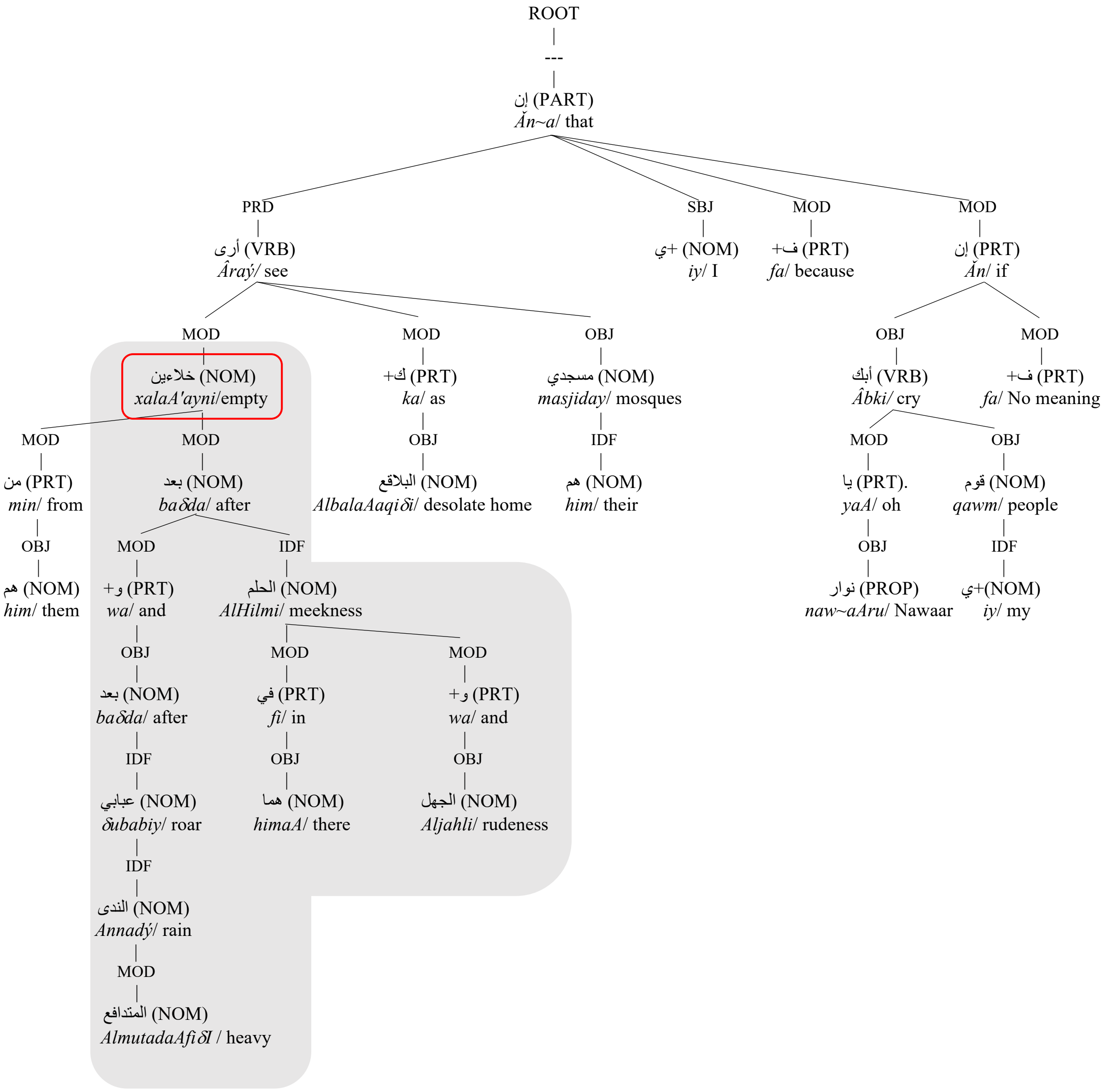


Figure 5. Dependency tree for two verses with dual syntactic relations

5.2 Broken and Complex Structure

As mentioned earlier in this paper, the selected poems were transmitted from an earlier era, using ancient CA. Since then, Arabic books have been published for each poet to collect and interpret their poems which guide the annotators during the manual verification work. These references show that some transmitted verses are broken, with missing parts or words. Also, some poems were incomplete.

For example, the poem starts with a verse that should be dependent on another unavailable previous verse. Therefore, broken and incomplete verses have been excluded from the corpus.

Although most of the related verses were sequential, we found more complicated cases that brought us to the alignment step in the annotation process. For example, the two contiguous verses (example 4 and 5) shown in Figure 5 have dual relations in both directions. Each verse includes a token headed by a parent token in the other verse. To illustrate the relations, we shaded all tokens in the dependency tree for the verse example 5. Its first word (خلاءين/ xalaA'ayni / empty), bordered by a red line, headed by a token and it heads another token that both are in the verse example 4. Placing the two verses in one syntactic tree shows the full structure that cannot be represented by an individual tree for each verse.

6 Evaluation

To test the effectiveness of the proposed annotations, we carried out some parsing experiments using dependency parsing models that adapted two different neural-based architectures. They achieved remarkable accuracies in dependency parsing for multilingual treebanks. The first model is the novel left-to-right dependency parser based on pointer networks developed by Fernández-González and Gómez-Rodríguez (2019). The second is the accurate and straightforward sequence tagging parser for Vacareanu et al. (2020).

We have split the ArPoT v1.0 randomly, dedicating 80% of the dataset for training. Due to the small size of the treebank and for a more confident result, 12% was used for testing and 8% for development. Words in this version are without “التشكيل/ *Taskeel*/ Diacritics”. We are planning to include them in the future. The treebank is available here: https://github.com/arpot-ksu.

Model	**Method**	**UAS**	**LAS**
Fernández-González and Gómez-Rodríguez (2019)	Transition based	81.52	75.25
Vacareanu et al. (2020)	Labeling	78.43	70.95

Table 2: Evaluation results on the ArPoT 1.0 test set for the two neural-based parsing models[4].

The parsing results are found in Table 2. We used the standard metrics for dependency parsing, Labelled Attachment Score (LAS) and Unlabeled Attachment Score (UAS). The reported scores are the average of three runs.

The accuracy of the transition-based pointer networks model is UAS of 81.52% and LAS of 75.25%, whereas the tagging model obtains a UAS of 78.43% and LAS of 70.95%. Overall, the results are promising for small treebank such as ArPoT. However, a more in-depth error analysis would be necessary to better understand the challenges of parsing models and provide an accurate analysis of CA poetry.

7 Conclusion and Future Work

This work described the first syntactically annotated corpus for Classical Arabic poetry. The treebank consists of 35,460 tokens. In addition to the annotation process, this paper discussed some issues during the development of the ArPoT treebank. We also posed an initial set of experiments with two neural-based parsing systems that show the appropriate settings of our treebank.

Future work plans will include more verses in our treebank and conduct a comparison study with other MSA treebanks. Also, we intend to further investigate the dependency parsing approaches on CA poetry. Besides, ArPoT might help in building a sentence boundary detection tool, which would be beneficial in our research.

[4] For both parsers we used the predefined settings.

Reference

Ahmed, Munef Abdullah, and Stefan Trausan-Matu. 2017. Using Natural Language Processing for Analyzing Arabic Poetry Rhythm. In *16th Networking in Education and Research RoEduNet International Conference,* 1–5, Targu-Mures, Romania. IEEE.

Albogamy, Fahad, Allan Ramsay, and Hanady Ahmed. 2017. Arabic tweets treebanking and parsing: A bootstrapping approach. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pp. 94-99.

Cohen, Jacob. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement,* 20 (1):37-46.

Dukes, Kais, and Nizar Habash. 2011. One-step statistical parsing of hybrid dependency-constituency syntactic representations. In *Proceedings of the 12th International Conference on Parsing Technologies*, pp. 92-103.

Dukes, Kais, and Tim Buckwalter. 2010. A dependency treebank of the Quran using traditional Arabic grammar. In *2010 the 7th International Conference on Informatics and Systems (INFOS)*, pp. 1-7. IEEE.

Fernández-González, Daniel, and Carlos Gómez-Rodríguez. 2019. Left-to-Right Dependency Parsing with Pointer Networks. 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),* pp. 710-716.

Habash, Nizar, Abdelhadi Soudi, and Timothy Buckwalter. 2007. On Arabic transliteration. In *Arabic computational morphology*, pp. 15-22. Springer, Dordrecht.

Habash, Nizar, Reem Faraj, and Ryan Roth. 2009. Syntactic Annotation in the Columbia Arabic Treebank. In *2nd International Conference on Arabic Language Resources and Tools MEDAR*, Cairo, Egypt.

Hajic, Jan, Otakar Smrz, Petr Zemánek, Jan Šnaidauf, and Emanuel Beška. 2004. Prague Arabic Dependency Treebank: Development in Data and Tools. In *The NEMLAR International Conference on Arabic Language Resources and Tools*, 110–117.

Javed, Talha, Nizar Habash, and Dima Taji. 2018. Palmyra: A platform independent dependency annotation tool for morphologically rich languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).*

Maamouri, Mohamed, Ann Bies, Seth Kulick, Michael Ciul, Nizar Habash, and Ramy Eskander. 2014. Developing an Egyptian Arabic Treebank: Impact of Dialectal Morphology on Annotation and Tool Development. In *LREC*, pp. 2348-2354.

Maamouri, Mohamed, Ann Bies, Tim Buckwalter, Mona Diab, Nizar Habash, Owen Rambow, and Dalila Tabessi. 2006. Developing and using a pilot dialectal Arabic treebank. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation* (LREC'06).

Maamouri, Mohamed, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR conference on Arabic language resources and tools*, Vol. 27, 466-467.

Ma, Xuezhe, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard Hovy. 2018. Stack-Pointer Networks for Dependency Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* pp. 1403-1414.

Najjar, Manal. 2012. Poetic Necessity Between the Syntax of a Sentence and the Syntax of a Text. *GSTF Journal of Law and Social Sciences (JLSS),* 2(1):322.

Shahrour, Anas, Salam Khalifa, Dima Taji, and Nizar Habash. 2016. Camelparser: A system for Arabic syntactic analysis and morphological disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pp. 228-232.

Suleiman, Saleh M 1990. The semantic functions of object deletion in classical Arabic. *Language Sciences*, 12(2-3): 255-266

Taji, Dima, and Nizar Habash. 2020. PALMYRA 2.0: A Configurable Multilingual Platform Independent Tool for Morphology and Syntax Annotation. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pp. 168-177.

Taji, Dima, Nizar Habash, and Daniel Zeman. 2017. Universal dependencies for Arabic. In Proceedings of the *Third Arabic Natural Language Processing Workshop*, pp. 166-176.

Vacareanu, Robert, George Caique Gouveia Barbosa, Marco A. Valenzuela-Escarcega, and Mihai Surdeanu. 2020. Parsing as tagging. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 5225-5231.

Zwettler, Michael. 1978. *Oral tradition of classical Arabic poetry: its character and implications*. The Ohio State University Press.

On auxiliary verb in Universal Dependencies: untangling the issue and proposing a systematized annotation strategy

Magali Sanches Duran
Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (USP)
magali.duran@uol.com.br

Adriana Silvina Pagano
Faculdade de Letras, Universidade Federal de Minas Gerais (UFMG),
apagano@ufmg.br

Amanda Pontes Rassi
Redação Nota 1000 Serviços educacionais / Somos Educação
amanda.rassi@somoseducacao.com.br,

Thiago Alexandre Salgueiro Pardo[1]
Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (USP)
taspardo@icmc.usp.br

Abstract

Auxiliary verbs are universally recognized as components of verbal constructions. While there is no shortage of scholarship on these verbs in various linguistic traditions, uncertainty still remains on the best way to annotate them for Natural Language Processing (NLP) purposes. This paper reviews the evolution of the concept of auxiliary verbs to gather insights into forms of representing them in an annotation scheme and raises some issues with a view to leveraging the potential afforded by them in different NLP tasks. Using Brazilian Portuguese as an instance language and Universal Dependencies (UD) as annotation model, we argue for (i) annotating inflected verbs as heads, (ii) annotating auxiliary interdependence in an auxiliation chain; and (iii) adopting a more consistent treatment of auxiliaries to encompass tense, aspect, modality and voice in auxiliation chains. We further propose auxiliary type as a feature to be annotated which can be easily implemented in existing and new treebanks with substantial gains in enriching the information that can be extracted for different NLP applications.

1 Introduction

Thousands of years ago, writing ushered in a new era for mankind. The advent of writing made it possible for thoughts and information to be conveyed between individuals across distinct epochs and localities. The ideas recorded in writing began to fertilize other minds and generate new ideas, exponentially accelerating the evolution of ideas in human societies (Ridley, 2010).

As much as writing allowed knowledge transfer among individuals, it today supports the transfer of human knowledge to machines. This happens through Natural Language Processing (NLP), and one of the ways to train machines to process text and learn to extract from text much of what humans do with it is through annotated corpora. Corpus annotation has thus become a way to formally record the implicit and explicit linguistic knowledge that can be gathered from texts. The result enables NLP to develop statistical models for training new human language technologies (Ide, 2017).

Annotating a corpus is an undertaking that requires considerable effort in designing, executing, and reviewing an entire process. Since design involves the creation of models to represent linguistic information, the models are often reused in corpus annotation endeavors, both within a language and in languages other than the one for which the model was created.

One drawback of leveraging models is the fact that, because they represent facts of a language, they are to some degree dependent on the language the theory drew on for its study. The advantage of models' reuse, on the other hand, is that the use of the same annotation scheme becomes a means to compare languages. The comparison, in turn, makes it possible to create multilingual NLP applications. This is the aim of the Universal Dependencies (UD) model (Nivre 2015, Nivre 2020), which is designed to be language independent. At the time of writing, there are over 200 corpora annotated with the UD model in just over 120 languages.

The fact that many linguists and computer scientists use UD and strive to instantiate it in their languages has promoted numerous discussions around its guidelines. One such discussion revolves around auxiliary verbs. Some languages, for instance, have opted for tagging tense and passive voice auxiliaries as AUX; others include modal verbs under this tag, and still others add aspectual verbs to the set. While UD does not require languages to follow a single standard, it recommends that only strongly grammaticalized auxiliary verbs be annotated as such. This, in turn, raises further discussion as to where to draw the line for an auxiliary to be considered fully grammaticalized.

This paper grew out of a concern on how to best represent auxiliary verbs in UD scheme towards building a proposal to leverage the full potential afforded by them in different NLP tasks. Drawing on Brazilian Portuguese, we contend that there are substantial gains to be obtained from (i) annotating auxiliaries as heads; (ii) annotating auxiliary interdependence in an auxiliation chain; and (iii) adopting a more consistent treatment of auxiliaries to encompass tense, aspect, modality and voice in auxiliation chains. We further propose auxiliary type as a feature to be annotated for the purpose of enriching information to be tapped from treebanks. Our proposal offers several benefits to NLP tasks, such as enhancing detection of subjects for information extraction by annotating inflected verbs as heads; temporal reference detection by relying on tense and aspect auxiliaries; and speculation detection by leveraging modal auxiliaries as cues for that task.

In Section 2, we briefly review the evolution of the concept of auxiliary verbs, highlighting the points that are important to our discussion. Section 3 discusses the auxiliation process and provides examples in Brazilian Portuguese for four types of auxiliaries: tense, aspect, modality and passive voice diathesis. In Section 4, we exemplify ways to annotate auxiliary verbs in UD, discussing their pros and cons and presenting a proposal to reduce different forms of annotation to an interpretation of auxiliaries common to all of them. Section 5 concludes our study.

2 Contributions over time towards the concept of auxiliary verbs

The topic of auxiliary verbs has been extensively discussed in the last decades, a thorough review being outside the scope of this paper. We will focus hence on the works that most contributed to advancing discussions of the auxiliary verb concept, since this concept is fundamental to the decisions about the UD annotation scheme we would like to argue for.

Our review begins with Tesnière (1959), an author who compared auxiliaries to free morphemes, though ascribed to them a distinctive nature of being inflectional. Auxiliary verbs, for Tesnière, help other verbs enable a subcategory transfer (in his account, of tense and voice) and are totally devoid of semantic content. Auxiliaries are classified as compound verb forms, as opposed to simple forms, and operate with auxiliated ones. Tesnière described auxiliary verbs as those assuming grammatical functions whereas auxiliated verbs contribute with the semantics. Being acquainted with English grammar (he mentions the verb *do* as an auxiliary), he admitted other functions of auxiliaries, which he suggested when he used *etc*. in: "One distinguishes between auxiliaries of tense (past, future), auxiliaries of voice (passive), etc." (Tesnière, 1959, p. 403).

Benveniste (1974) elaborated on Tesniere's description, recognizing verbal chains of auxiliaries of tense, modality and diathesis (voice). The author coined the term *auxiliation* to refer to a process that syntagmatically joins an auxiliating form to an auxiliated one, avoiding the use of the term *auxiliary*. For simplicity, we adopt the term *auxiliary verb*, even when referring to Benveniste's work.

By including modal verbs as a further type of auxiliation, Benveniste evidenced that auxiliation chains are longer and more complex than previously believed; yet he did not include aspectual verbs among auxiliaries. Another important contribution by Benveniste was to show that there is sequential order for auxiliary verbs to occur, namely, modal - temporal - passive voice - full verb, within a process he called *over-auxiliation*. Despite acknowledging that an auxiliary verb is the verb that takes person, number, mood, and tense inflections in a compound form and showing compound forms made up by up to three types of auxiliary functions (tense, modality, and passive voice diathesis), Benveniste did not expand on the fact that, in the case of longer chains, auxiliaries after the first one do not take inflections. Neither did he explicitly state that the second auxiliary in a chain is auxiliated by the first one and so on. He did state, however, that an auxiliary of passive voice diathesis is always the last one in a chain before a full verb, because no auxiliary verb can undergo passive voice diathesis.

Another relevant contribution to the concept of auxiliary verbs was made by studies of grammaticalization, mainly after the 1990s. Heine (1993) does a survey of the different ways languages express features of tense, modality and verbal aspect, pointing out that the lack of agreement around a concept of auxiliaries is largely due to the diversity of phenomena. For Heine, one of the sources for linguists' disagreement can be traced to Chomsky's AUX, a universal category he introduced in 1956, which is in fact not directly related to auxiliary verbs. Heine's review shows that auxiliary verbs have been at times considered as main verbs, as non-autonomous verbs, and still as a different grammatical category of verb altogether. Likewise, in dependency grammars, auxiliary verbs are usually considered as dependent by some authors while others posit them as heads. Steele (1994, p.818) praised Heine's work for his survey of views on auxiliary verbs, but criticized him for not tackling issues such as which verb is head and which one is dependent in dependency relations.

Kuteva (2001), who set out to complete the work of Heine, remarked that the big problem is the fact that some linguistic traditions disregard the dynamic character of the process of auxiliation, which prevents new auxiliaries arising in languages from being recognized. For her, auxiliation is an ongoing process and auxiliary verbs can be found at various stages in this process. There is thus no limited set of auxiliary verbs and one cannot separate auxiliary verbs from the verbs that gave rise to them.

Andersen (2006) agrees that auxiliation is a dynamic process, “so the class is continually losing and acquiring new members” (p.4). The author compares auxiliary constructions in over 800 languages and concludes: “There is no, and probably cannot be, any specific, language independent formal criteria that can be used to determine the characterization of any given element as a lexical verb or an auxiliary verb.” (p.5). He makes an important distinction between inflectional and semantic heads. The former encodes features responsible for making the construction to be grammatical, whereas the latter determines valence (argument structure). In some languages, the inflectional and the semantic heads are conflated, as the full verb is the one bearing inflections. In others, the inflectional head is the auxiliary and the semantic head is the full verb. Therefore, depending on the annotation purpose, dependence relations may prioritize the inflectional or the semantic head[1].

Krug (2012) discusses the grammaticalization of auxiliary verbs and illustrates the process of full verbs becoming auxiliaries for the English language. According to the author, and for this he draws on Bolinger (1980, apud Krug, 2012), it suffices for a verb to receive a complement in infinitive form to enter a path of grammaticalization. Krug cites the following characteristics of auxiliaries:

- they may coexist with a full homonymous verb;
- they contribute to expressing tense, aspect and modality (known as TAM);
- they do not occur alone, except in cases of elliptical full verbs (easily recoverable in context);
- they are complemented by verbs in non-finite forms (gerund, participle and infinitive).

[1] In fact, Andersen (2006) basically recognizes three patterns of auxiliary verb construction inflections: AUX-headed (the auxiliary is the inflected verb), which is the most common pattern; LEX-headed (the full verb is the inflected verb, as in Eneats, Bulgarian, Macedonian, Hatam, Koiari and Kwerba); and doubled inflections (both auxiliary and full verbs inflect, as in Gutob, Mombelo and Mumbami).

Krug acknowledges the fact that auxiliaries for passive voice diathesis and negative and interrogative constructions are considered by some linguists, but he does not include them in his account.

3 Discussion

Prior to Benveniste, an auxiliary construction was seen as a set of verbs complementing each other: one of them expressing grammatical features and the other semantic ones. Benveniste posited *over-auxiliation* and included modal verbs in the auxiliation process. The chain of over-auxiliation, including aspectual verbs, conforms to the following sequence of occurrence: modal - temporal - aspectual - passive voice - full verb (see Figure 1). Over-auxiliation is of key importance for corpus annotation, though it is still under-explored in accounts on the matter. Bearing in mind that each auxiliary verb imposes a non-finite form on the auxiliated verb, we can argue that, except for the first auxiliary, which holds inflections, all the other auxiliaries in a verbal chain are, concomitantly, auxiliated by a preceding verb and auxiliary to the following one, as shown in Figure 1. This implicates that the traditional labor division ascribed to auxiliary verbs as representing grammatical functions and full verbs as representing semantic functions cannot be sustained.

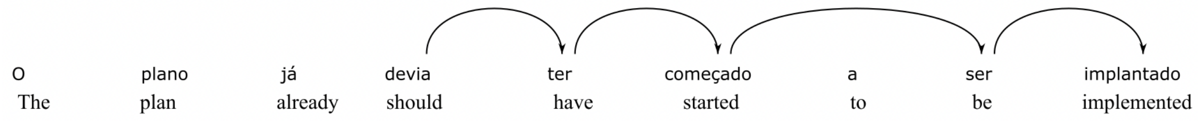


Figure 1: Auxiliary chain showing over-auxiliation process

Verbs in Figure 1 can be thus analysed:

- **devia [dever (should)]**, auxiliary of modality, requires the auxiliated to be an infinitive form; therefore, the verb auxiliated by *devia* is the verb *ter*.
- **ter [ter (have)]**, auxiliary of tense, requires the auxiliated to be a past participle form; the verb auxiliated by *ter* is the verb *começar*. *ter* is auxiliated by *devia* and is auxiliary to *começar*.
- **começado [começar (start)]**, auxiliary of aspect, requires the auxiliated to be an infinitive form and to be introduced by the preposition *a*; the verb auxiliated by *começar* is the verb *ser*. Therefore, *começar* is auxiliated by *ter* and is auxiliary to *ser*.
- **ser [ser (be)]**, auxiliary of passive voice, requires the auxiliated to be a past participle form. The verb auxiliated by *ser* is the verb *implantar* (past participle: *implantado*), which is the full verb in this sentence. Therefore, *ser* is auxiliated by *começar* and auxiliary to *implantar*.
- **implantado [implantar (implement)]** is a full verb, auxiliated by *ser*.

A productive way to explore the concept of auxiliation is to focus on the concept of auxiliated verb rather than on the concept of auxiliary verb. An auxiliated verb may be an auxiliary or a full verb. An auxiliated verb is a verb that takes the non-finite form required by its auxiliary, is introduced by the preposition (if any) required by its auxiliary, and has the same subject as its auxiliary. Therefore, we may have an auxiliation chain whenever all verbs in a chain share the same subject. However, we cannot affirm that the verbs in a chain sharing the same subject are auxiliaries followed by a full verb, as it depends on which verbs will be considered auxiliary in each work and for what purpose. In auxiliation verbal chains, the first verb is only auxiliary and the last verb is only auxiliated (full verb), but the verbs in between are both auxiliary and auxiliated, which shows that these two categories are not mutually exclusive.

The question to be posed is not whether a verb is an auxiliary, but whether it is an auxiliary in a given chain and what can be leveraged from its annotation. This approach makes it possible to overcome the much debated need to define a list of auxiliary verbs. Discussions about whether a verb is an auxiliary or not have always been based on comparing verbs with prototypical auxiliaries, i.e., those whose grammaticalization process is well advanced. In English, some auxiliaries (*can*, *may*, *might*, *should*) are fully grammaticalized to the extent that they do not compete with homonymous full verbs, do not require *to* to introduce the auxiliated verb, and do not require another verb to construct a negative and an interrogative form[2]. In other languages, such as Brazilian Portuguese, auxiliary verbs are at different stages in the grammaticalization process.

Several scholars have tackled the task of describing auxiliary verbs in Brazilian Portuguese. Among them, Pontes (1973) and Lobato (1975) are two particularly exhaustive accounts, each exploring likely criteria to classify a verb as an auxiliary one. Pontes (1973) takes a syntagmatic view on auxiliaries and discusses interdependence relations between auxiliary and auxiliated verbs. Lobato (1975) performs different probes to try to differentiate auxiliary from non-auxiliary verbs. More recently, Ilari & Basso (2014) systematize criteria to assign auxiliary status to a given verb, considering all instances found in corpora regardless of the degree of grammaticalization a verb is still exhibiting. When it comes to grammar textbooks, lists of auxiliary verbs can be found in most of them, no two lists being alike, which shows the variety of criteria and stances taken by grammarians in Brazil.

Drawing on Portuguese as a sample case, we would like to argue for a view on auxiliary verbs as verbs in their own right, implicating that a verb can be an auxiliary verb in some uses and a full verb in others, in the latter operating to help construe a variety of meanings. While they may be semantically weak, auxiliary verbs have a strong role in the syntax of the clause, its finite form agreeing with the subject and dictating the form of their auxiliated verbs. Auxiliary verbs can take part in a chain of several auxiliary verbs and be modified by adverbs, which sets them apart from fully-grammatical words. To better grasp the behavior of each auxiliary type in Brazilian Portuguese, we will briefly address four main groups (tense, modality, aspect and passive voice diathesis) and their characteristics below.

3.1 Auxiliaries of tense

Brazilian Portuguese expresses tense basically through morphological desinence. The so-called tense auxiliaries *ter* and *haver* are used to express a previous past event within the past itself (1) and a previous future event within the future (2). For this reason, *ter* and *haver* are tense auxiliaries in some environments only, as in the following examples.

(1) Quando olhei, ele já **havia atirado**. (When I looked, he **had** already **shot**.)
(2) No dia que você vier eu já **terei partido**. (The day you will arrive I **will be gone**.)

As seen above, both *ter* and *haver* require the auxiliated verb to be a past participle form. However, a past participle form is not a criterion sufficiently strong to single out occurrences of *ter* and *haver* as tense auxiliaries. *Ter* may combine with a past participle form in other tenses to express aspect (3) and resultative constructions[3] (4).

(3) Ele **tem vindo** aqui todos os dias. (He **has been coming** here every day.)
(4) Ele **teve aprovado** seu visto só ontem. (He **had** his visa **approved** only yesterday.)

In the case of *haver*, this verb may be followed by a past participle form, which is not actually a verb, but a noun. In such cases, *haver* is not an auxiliary verb. This is the case of (5) and (6):

(5)	Não	**houve**	**comunicado**	prévio	dos	organizadores
	Not	had	communicated[4]	prior	of the	organizers

There was no prior notification by the organizers.

[2] Among less grammaticalized auxiliaries in English, Osborne & Gerdes (2019) point out *be going to.*

[3] Resultative constructions resemble a kind of diathesis where the subject has the semantic role of benefactive. Diathesis is the alternation of semantic roles: in the passive voice diathesis, the patient is the subject; in the causative diathesis, the cause is the subject.

[4] The word *comunicado* is the past participle of the verb *to communicate* and means both *communicated* and *notification*. Many past participles in Portuguese are employed as true nominals.

(6)	Não	**há**	**sentido**	em	fazer	isso
	Not	have	felt[5]	in	to do	this
	There is no sense in doing that.					

In examples 5 and 6, *comunicado* and *sentido* are nouns and not verbs. In both, *haver* construes existence and is inflected in present and perfect tenses, i.e., it is not an auxiliary of tense.

3.2 Auxiliaries of modality

Brazilian Portuguese has two modal verbs that are more highly grammaticalized than others: *poder* and *dever*. Both express several types of modality: permission, obligation and possibility. *Poder* has no homonymous full verb, but *dever* does [*dever* (*to owe*)]. There are several less grammaticalized modals like, e.g., *tentar* (*to try*). Some of them can be used as full verbs as is the case of *saber* (*to know*) and some of them, as is the case of *pretender* (*to intend*) and *querer* (*to want* or *would like to*), can also take a finite clause as a complement, its subject not being the same as the one of the main clause. However, whenever followed by an infinitive, those verbs share the same subject. These two possibilities are illustrated by examples (7) and (8).

(7) Você **quer marcar** uma consulta semana que vem? (**Would you like** to **schedule** an appointment next week?)

(8) Você **quer** que eu **marque** uma consulta semana que vem? (**Would you like** me to **schedule** an appointment next week?)

The particular behaviour of modal verbs mentioned above tends to exclude such verbs from traditional lists of modal verbs in Portuguese[6]. However, for NLP, verbs like *pretender* (to intend), *querer* (to want), *saber* (to know), *tentar* (to try), etc. followed by an infinitive verb are important cues for deducing whether an event has occurred or whether a statement is a fact or mere speculation.

Modal verbs in Brazilian Portuguese are in a stage of grammaticalization in which they have not lost their semantic load, since even the most grammaticalized one, *poder*, is polysemous: it construes permission (9) or probability (10). The same holds for *dever*, which construes obligation (11) or probability (12):

(9) Você **pode entrar**, se quiser. (You **may come** in if you want.)

(10) **Pode chover** hoje à noite. (It **may rain** tonight.)

(11) O funcionário **deve usar** uniforme todos os dias no trabalho. (Employees **must wear** their uniform every day at work.)

(12) O atraso **deve ser** por causa da chuva. (The delay **must be** due to the rain.)

Grammaticalization studies point out that there may be verbs at various stages in the grammaticalization process regarding their use as auxiliaries and this seems to be the case for many modal verbs in Brazilian Portuguese.

3.3 Auxiliaries of Aspect

Aspectual verbs express how events occur in time. The meaning of aspect can be readily grasped through examples of some of its subcategories: frequentative (informing an event repeats frequently), inchoative (informing an event has started) and terminative (informing an event has finished).

[5] The word *sentido* is the past participle of the verb sentir (*to feel* and means both *felt* and *sense*.

[6] In fact, some modal verbs such as *querer*, *desejar*, *pretender* are classified as full verbs realizing mental processes in systemic-functional descriptions of Portuguese. Likewise, in some accounts on auxiliary verbs, modals do not fulfill all the criteria to be considered auxiliary verbs (cf. LOBATO, 1975).

There are aspectual verbs in Brazilian Portuguese that convey information on the event. For example, the aspectual verb *chegar a* (literally *arrive to*) followed by an infinitive signals the event took place some time ago and lasted for a while, but did not persist.

(13)	Ele	**chegou**	**a**	**pensar**	em	abandonar	o	Brasil
	He	arrived	to	think	in	to abandon	the	Brazil

He **even thought** about leaving Brazil for good.

If we did not take into account the dynamic character of the process of auxiliation, as pointed out by Kuteva (2001), we would not be able to recognize new aspectual verbs arising in Portuguese. For example, the verb *dar de* [*dar* (give)], followed by infinitive, informs the event has become a habit:

(14)	Ele	**deu**	**de**	**assistir**	filmes	de	terror	ultimamente
	He	gave	of	watch	movies	of	horror	lately

He **took to watching** horror movies lately.

(15)	Eu	**dei**	**de**	**suspeitar**	de	todo	mundo	depois	que	fui	enganada
	I	gave	of	suspect	of	all	world	after	that	was	deceived

I **became suspicious** of everyone after I was deceived.

Besides being prolific, aspectual verbs are the least grammaticalized verbs in Brazilian Portuguese. Some compete in interpretation with full verbs, as is the case of *acabar de* [*acabar* (finish)], which, followed by an infinitive introduced by the preposition *de*[7], is aspectual in (16) and full verb in (17).

(16)	O	filme	**acabou**	**de**	**começar**
	The	movie	finished	of	to start

The movie **has just started**.

(17) Ele já **acabou de ler** o livro. (He has already finished reading the book.)

3.4 Auxiliary of passive voice

In Portuguese, the passive voice may be constructed by the auxiliary verb *ser* (to be) followed by a past participle (which we call analytic passive voice) or by adding the pronoun *se* to a transitive verb (which we call synthetic passive voice). In analytic passive voice, the subject is the prototypical patient and comes to the right of the verb *ser*:

(18) As cartas de sentença **foram assinadas** pelo juiz. (The sentencing letters **were signed** by the judge.)

Unlike the auxiliary verbs of tense, which also require the participle form of their auxiliated, the participle of the passive voice is not invariable: it agrees in number and gender with the subject of the passive voice (in Portuguese, number and gender are typical inflections of nouns, while the typical inflections of verbs are mood, person, number, and tense). This fact enables us to verify agreement between verb and subject regardless of which verb is the head of the subject dependency relation.

In Portuguese, only the passive voice auxiliary is fully grammaticalized and may occur in all verb tenses. Tense auxiliaries are well grammaticalized, but need to be annotated as such, though only in some verb tenses. Modals and aspectual verbs are less grammaticalized, but this does not mean their annotation is less important for NLP.

[7] One feature of aspectual verbs that poses a challenge to their annotation as auxiliaries in UD is the fact that most of them require a preposition to introduce the auxiliated verb. UD does not provide a specific dependence relation to link this preposition to one of the verbs (since the preposition neither marks case, nor introduces a subordinate clause). We have opted for annotating prepositions for verb and noun arguments as ADP.

4 Auxiliaries in Natural Language Processing

The different views among linguists on which verb categories can actually be considered auxiliary pose an additional challenge to model them for the purpose of annotation in NLP.

For syntactic annotation based on constituencies, the key issue is to decide which of the verbs in a verbal phrase is the head. For annotation based on dependencies, the decision encompasses which verb is head and which one is dependent, as well as which dependency relation links each of the verbs to the others.

In UD, there is a Part of Speech (Pos) tag for full verbs (VERB) and a PoS tag for auxiliary verbs (AUX). Its guidelines leave it up to each language to define which categories of auxiliaries will be annotated as AUX and which verbs are prototypically used in each category. UD basically recommends that the auxiliary verbs specified in the annotation guidelines should be highly grammaticalized in the language. This has encouraged very conservative decisions, so that not all languages annotate modality auxiliaries (modal verbs) and aspect auxiliaries (aspectual verbs), as they tend to be less grammaticalized than other auxiliaries.

Automatic identification of auxiliaries in Portuguese has already been focused on by Baptista et al. (2010), who consider an extensive list of over 26 auxiliary verbs; however, despite aiming at a dependency parser, the authors do not follow UD guidelines.

In Portuguese, one of the main probes for identifying the subject is through verb agreement. The verb holding verb inflections is thus a natural candidate to be the head of the **nsubj** relation. This has implications in cases where there are auxiliaries together with a full verb. In UD, once a verb is annotated as AUX, there is no possibility to annotate it as a head: it has to be dependent in an **aux** dependency relation. Only verbs annotated as VERB may be heads of dependency relations.

Therefore, when an auxiliary verb, annotated as AUX, happens to be the first in an auxilation chain, thus keeping the inflections, a direct dependency relation between the inflected verb and the subject is not annotated, which precludes easy extraction of subjects. Still, if a verbal auxiliation chain contains several verbs annotated as AUX, the distance between the subject and the head (full verb) is longer and, as the full verb is always in an infinite verb form, the main cue to verify agreement between subject and verb can be missed. Most importantly, once a verb annotated as AUX cannot be head in a dependency relation, this prevents annotating over-auxiliation, which is marked by the non-finite verb form required from an auxiliated by its auxiliary. This is a major drawback, implicating that rich morphological and syntactical information is left untapped for linguistic studies and NLP applications.

As far as we can see, there are three options to annotate auxiliation chains using dependency relations, which are illustrated in the three figures below.

Figure 2 shows annotation of all verbs with the PoS tag VERB, the first one being the head and the second one a dependent in a **xcomp** dependency relation, this annotation being iterated between the following verbs in the chain.

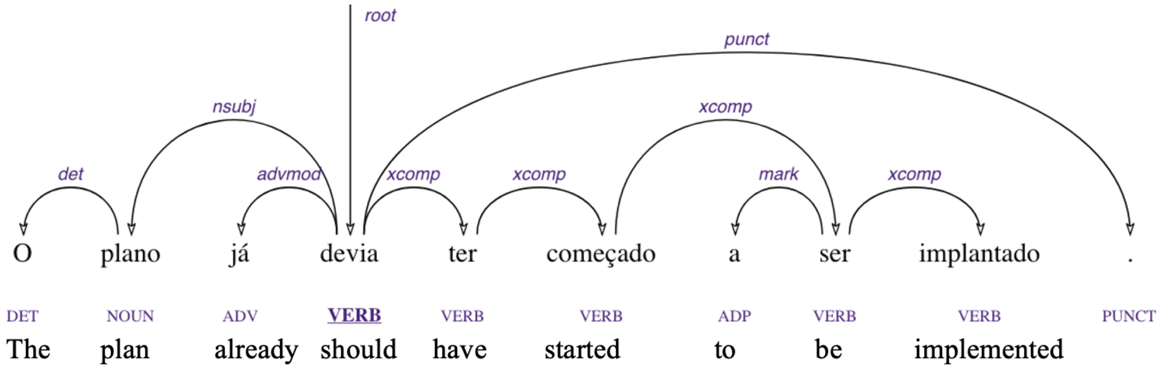


Figure 2: Annotation of inflected verb (auxiliary) as head

Figure 3 shows annotation of the more grammaticalized verbs as AUX (tense and passive voice auxiliaries) and the less grammaticalized ones as VERB (modal and aspectual auxiliaries), obtaining a combination of dependency relations **aux** and **xcomp**.

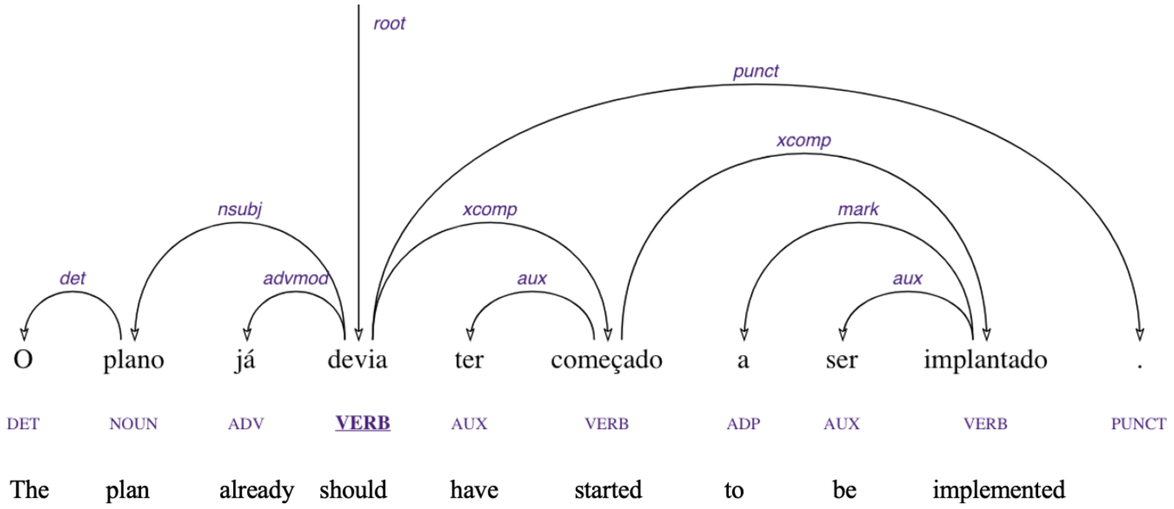

root
punct
xcomp
nsubj
xcomp
mark
det
advmod
aux
aux
O plano já devia ter começado a ser implantado .
DET NOUN ADV VERB AUX VERB ADP AUX VERB PUNCT
The plan already should have started to be implemented

Figure 3: Hybrid annotation of auxiliaries and auxiliated verbs

Figure 4 shows annotation of all verbs as AUX, except for the last one, which is a full verb and takes the root.

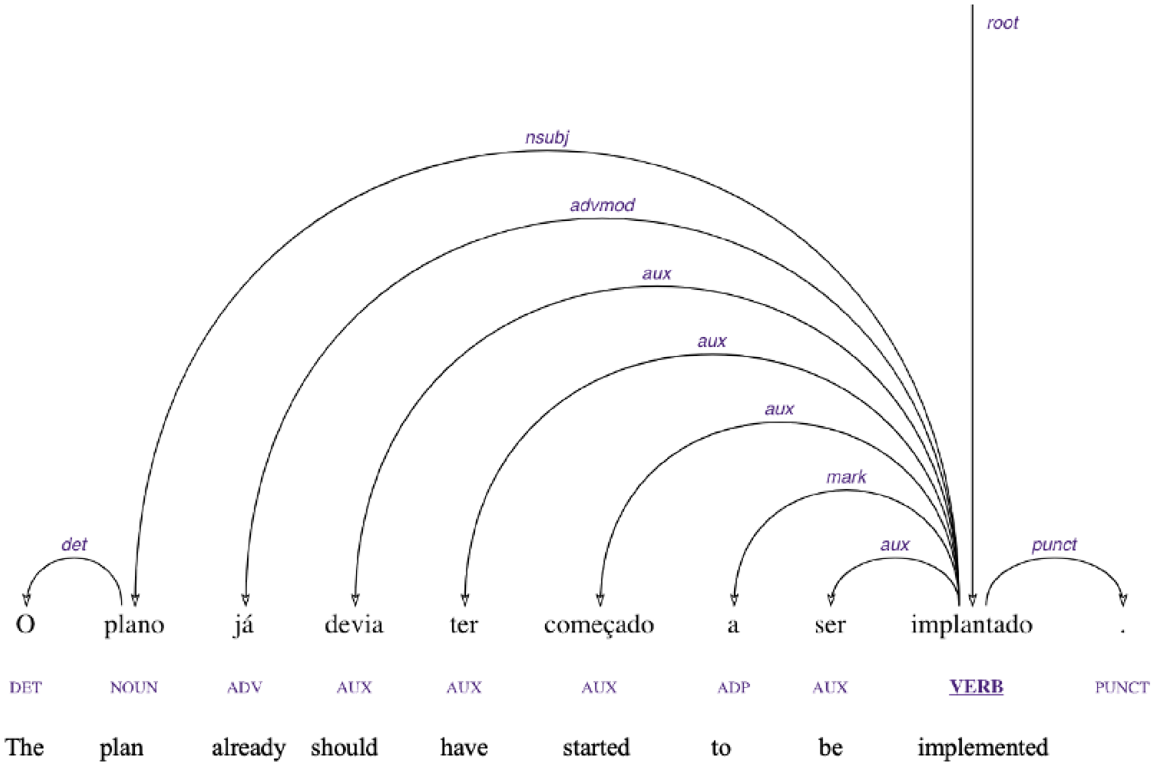

root
nsubj
advmod
aux
aux
aux
mark
det
aux
punct
O plano já devia ter começado a ser implantado .
DET NOUN ADV AUX AUX AUX ADP AUX VERB PUNCT
The plan already should have started to be implemented

Figure 4: Annotation of (auxiliated) full verb as syntactic head

Annotation in Figure 2 is the most satisfactory as it tags all verbs alike, which is a better representation in the case of a chain, where one verb is concomitantly auxiliary to the following verb and auxiliated by the preceding one. Annotation in Figure 3 is less satisfactory; although it keeps the traditional annotation of more grammaticalized auxiliary verbs as AUX, it misses details regarding the syntactic relation between the verbs annotated as VERB and those annotated as AUX, such as the requirement for an infinitive form after the aspectual verb *começado* (*started*). It also splits the verb chain into two xcomp relations. Annotation in Figure 4 is the least satisfactory of all, as it ignores the interdependence relationship between the verbs in a verbal chain and poses a problem regarding the distance between the verb holding inflections and the subject, assuming that the longer the distance is, the more difficult the task of subject detection becomes.

In order to find out which strategy is more suitable for machine learning, Lhoneux et al. (2020) compared two ways of annotating auxiliaries in UD: the auxiliary as head of its auxiliated (head left and dependent right) and the auxiliary as dependent on the auxiliated (head right and dependent left). They concluded that the information the annotation brings to the process depends on the machine learning modeling choices, and, therefore, if properly modeled, the same properties may be automatically acquired. Given that machine learning seems to deal equally well with both forms of annotation, we believe that it is preferable to choose a learning model based on what is desired to be learned rather than to choose a form of annotation based on what the available models can learn.

In Brazilian Portuguese auxiliary constructions, the inflectional head is an auxiliary and, for syntactic purposes, it is the head of the construction, as it must agree with the subject. In UD, however, when a verb is annotated as AUX, it becomes a dependent in a dependency relation **aux**, the head being another verb: a full verb or other auxiliary verb annotated as VERB if there is a chain. Moreover, UD defines AUX as a functional word and restricts its selection as head in a relation. This means that the phenomenon of over-auxiliation (one auxiliary modifying another) cannot be represented using AUX in UD.

From the perspective of semantic applications in NLP, treatment of auxiliaries and full verbs has important implications. Promoting the full verb to head is interesting to information extraction. Buiko et al (2009), for example, compared the effect of different dependency representations on information extraction and concluded that the *trimming* of auxiliary structures enhanced the event extraction results. The trimming of auxiliary structures is an operation that seeks to “prune the auxiliaries/modals as governors from the dependency graph and propagate the dependency relations of these nodes to the main verbs” (Buiko et al 2009). For temporal expression, aspectual and modal verbs are fundamental cues, as explained by Pustejovsky et al (2017) in their design of the TimeML model, aimed to extract time information on events. Modal verbs are also relevant cues for speculation detection, as explored for different domains in Ozgur & Radev (2009), Zhou et al. (2010), Sauri & Pustejovsky (2012), and Rivera Zavala & Martinez (2020).

Since an auxiliation chain has the same subject and refers to a same event, identifying them is productive for extraction tasks, even if auxiliaries are annotated as head or not, and even when there are other functions in between, such as adverbs and pronouns.

Three main arguments are worth summing up at this point:

- there is syntactic relationship between the auxiliaries of modality, tense, aspect and passive voice diathesis within a chain of over-auxiliation;
- verbs in a chain can be at the same time auxiliated by a verb and auxiliary to another one;
- the UD guidelines do not allow an auxiliary verb (considered a functional word) to be head of a dependency relation;

Bearing upon the above arguments, we believe the most productive way to annotate auxiliary verbs in UD is using the tag VERB and relating verbs to each other as open clausal complements (**xcomp**), this relation tag implicating they all have the same subject. Moreover, considering that modality, tense, aspectual and passive voice cues are relevant for many NLP applications, we propose to add a new annotation for this purpose at the morphological level: a feature called *AuxiliaryType*, with the initial values: Tense, Modality, Aspect, and Voice. Thus, regardless of whether the verbs that participate in the auxiliation chain have been annotated as AUX or as VERB, they will be identified at the feature level by the auxiliary function they perform within the chain. The absence of this feature means that the verb is not performing an auxiliary function within the chain and is therefore a full verb.

Our proposal has a twofold impact:

- it allows treebanks with other annotation decisions for auxiliaries to reconcile their annotations by simply adding a feature to indicate auxiliary type (no alteration in annotation needed, but merely addition of features);
- it enables recovery of auxiliation chains, as a sequence of verbs and/or auxiliaries that present a value of AuxiliaryType and are followed by a verb with no value of AuxiliaryType (the full verb, which is the semantic head).

A further proposal is specifying Voice at the feature level. UD provides features to discriminate categories of Modality (Mood), Tense and Aspect. However, it has no feature to discriminate between categories of Voice (values: Passive, Agentive, Resultative, Causative, etc.). It would therefore also be desirable to create a Voice feature to complete the description of auxiliaries in UD morphology. This is particularly interesting because Voice is directly linked to the semantic role of the subject (Patient, Agent, Beneficiary, Cause, etc.), and this would favor other NLP semantic applications.

5 Final Remarks

In this paper, we have discussed the annotation of auxiliary verbs under the UD model, using evidence from linguistic theory to reconcile different ways of annotating phenomena that share semantic similarity, but differ greatly in syntactic behaviour.

As we have argued in the preceding sections drawing on grammaticalization studies, auxiliary verbs are verbs in their own right, i.e., they are neither a closed class of words (they are open to new candidate forms to auxiliaries), nor are they a fully functional class of words (they can operate as full verbs themselves). Throughout this paper, we have put forward arguments in favour of annotating auxiliary verbs as heads in dependency relations whenever the inflected verb is an auxiliary verb, as in Portuguese auxiliary chains. Hence, our proposal is to leverage the role of auxiliary verbs in the syntax of the clause, both in determining the form of auxiliated verbs and establishing agreement with the subject.

We also proposed the inclusion of a new morphological feature, *AuxiliaryTyp*e, with the initial values of Tense, Modality, Aspect, and Voice. This way, we move towards standardizing UD auxiliaries annotation, enhancing comparability between languages. By annotating information on auxiliary function at feature level, we can reconcile our proposal to annotate auxiliaries with that of other languages that may have adopted other annotation strategies, either because their auxiliary verbs are not inflected, or because they have decided to privilege the semantic head in syntactic annotation.

Acknowledgements

Magali Duran and Thiago Pardo are grateful to The Center for Artificial Intelligence of the University of São Paulo (C4AI-http://c4ai.inova.usp.br/), sponsored by IBM and FAPESP (grant#2019/07665-4). Adriana Pagano wishes to thank the National Council for Scientific and Technological Development (CNPq) for grant No. 310630/2017-7.

References

Anderson, Gregory D. S. 2006. *Auxiliary Constructions*. Oxford University Press.

Baptista J., Mamede N., Gomes F. 2010. Auxiliary Verbs and Verbal Chains in European Portuguese. In: Pardo T.A.S., Branco A., Klautau A., Vieira R., de Lima V.L.S., eds. Computational Processing of the Portuguese Language. PROPOR 2010. *Lecture Notes in Computer Science*, vol 6001. Heidelberg: Springer. https://doi.org/10.1007/978-3-642-12320-7_14

Benveniste, Émile. 2006. Estrutura das Relações de Auxiliaridade. In: *Problemas de Linguística Geral II*. Campinas: Pontes, [1965], pp. 181-198.

Buyko, E., Faessler, E., Wermter, J., & Hahn, U. (2011). Syntactic simplification and semantic enrichment-trimming dependency graphs for event extraction. *Computational Intelligence*, 27(4), 610–644. doi:10.1111/j.1467-8640.2011.0

Heine, Bernd. 1993. *Auxiliaries: Cognitive Forces and Grammaticalization*. New York: Oxford University Press.

Ide, Nancy. 2017. Introduction: The Handbook of LinguisticAnnotation. In: Ide, Nancy; Pustejovsky, James, eds. *Handbook of Linguistic Annotation*. Springer. DOI 10.1007/978-94-024-0881-2_1

Ilari, Rodolfo, Basso, Mario. 2014. O verbo. In Ilari, Rodolfo. *Gramática do português culto falado no Brasil* - vol. III - palavras de classes abertas.

Krug, Manfred. 2012. Auxiliaries and grammaticalization. In: Heine, Bernd; Narrog, Heiko, eds. *The Oxford Handbook of Grammaticalization.* Oxford University Press. DOI: 10.1093/oxfordhb/9780199586783.013.0044

Kuteva, Tania. 2001. *Auxiliation: an enquiry into the nature of grammaticalization*. New York & Oxford: Oxford University Press.

Lhoneux, Miryam de, Sara Stymne, Joakim Nivre. 2020. What Should/Do/Can LSTMs Learn When Parsing Auxiliary Verb Constructions? *Computational Linguistics* 46(4), pp. 763-784.

Lobato, Lúcia M.P. 1975. A auxiliaridade em português. In: Lobato, Lúcia et al., ed. *Análises linguísticas*, pp. 27-91. Petrópolis: Vozes.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers & Daniel Zeman. 2020. Universal Dependencies v2: An ever growing multilingual treebank collection. In: *Proceedings of the12th Language Resources and Evaluation Conference*, 4034-4043. Marseille, France: European Language Resources Association. https://aclanthology.org/2020.lrec-1.497.

Nivre, Joakim. 2015. Towards a universal grammar for natural language processing. In: Alexander Gelbukh, ed. *Computational Linguistics and Intelligent Text Processing*, pp. 3–16. Cairo, Egypt: Springer International Publishing. doi:10.1007/978-3-319-18111-01.

Osborne, Timothy and Kim Gerdes. 2019. The status of function words in dependency grammar: A critique of Universal Dependencies (UD). *Glossa: a journal of general linguistics* 4(1):17. pp. 1–28, DOI: https://doi.org/10.5334/gjgl.537

Ozgur, Arzucan, Dragomir R. Radev. 2009. Detecting Speculations and their Scopes in Scientific Text. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 1398–1407, Singapore.

Pontes, Eunice. 1973. *Verbos Auxiliares em Português*. Rio de Janeiro, Ed. Vozes.

Pustejovsky, James; Bunt, Harry; Annie. Zaenen. 2017. DesigningAnnotation Schemes: From Theory to Model. In: Ide, Nancy; Pustejovsky, James, eds. *Handbook of Linguistic Annotation*. Springer. DOI 10.1007/978-94-024-0881-2_1

Ridley, Matt. 2010. *The Rational Optimist: How Prosperity Evolves*. HarperCollins Publishers.

Rivera Zavala R, Martinez P. 2020. The Impact of Pretrained Language Models on Negation and Speculation Detection in Cross-Lingual Medical Text: Comparative Study. *JMIR Med Inform* 8(12):e18953 URL: https://medinform.jmir.org/2020/12/e18953 DOI: 10.2196/18953

Sauri, Roser, Pustejovsky, James. 2010. Are You Sure That This Happened? Assessing the Factuality Degree of Events in Text. *Computational Linguistics*, 38, 2, pp. 261-299.

Steele, Susan. 1994. Reviewed Work: Auxiliaries: Cognitive Forces and Grammaticalization by Bernd Heine In: *Language*, Vol. 70, No. 4, pp. 818-821. Linguistic Society of America Language https://doi.org/10.2307/416332. Accessed August 13, 2021.

Tesnière, Lucien. 1959. *Elements of Structural Syntax*. Osborne, Timothy; Kahane, Sylvain, translators. John Benjamins Publishing Company, 2015.

Zhou, H., Li, X., Huang, D., Li, Z., & Yang, Y. 2010. Exploiting Multi-Features to Detect Hedges and their Scope in Biomedical Texts. In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning – Shared Task.* pp. 106–113. Uppsala: Association of Computational Linguistics.

How useful are Enhanced Universal Dependencies for semantic interpretation?

Jamie Y. Findlay
University of Oslo / Oslo, NO
`jamie.findlay@iln.uio.no`

Dag T. T. Haug
University of Oslo / Oslo, NO
`d.t.t.haug@ifikk.uio.no`

Abstract

We discuss the role of enhanced Universal Dependencies (E-UD) in the task of deriving semantic predicate-argument structures from UD treebanks in a universal, non-language-specific way. We consider the usefulness of three kinds of E-UD annotation (controllers of `xcomp`s, propagation of outgoing dependencies in coordinations, and coreference in relative clauses) and assess some heuristics for automatically adding such enhancements. We conclude that one large obstacle both for deriving predicate-argument structures from UD treebanks and for the automatic enhancement of basic UD treebanks is the fact that UD does not represent empty elements such as pro-dropped arguments, and we suggest that devoting effort to this would often provide a better return on investment than spending resources on improving or adding E-UD annotations.

1 Introduction

One important, traditional application of syntactic analysis is to support the creation of meaning representations. In fact, formal semantics in the tradition of Montague (Montague, 1970; Heim and Kratzer, 1998) holds that syntactic structure together with lexical meaning *determines* sentence meaning. But even if we do not accept that strong view, it is clear that syntax informs and constrains meaning. In particular, this is true of predicate-argument structures, which we take to involve relating each entity referred to in a sentence to an appropriate eventuality either directly or indirectly (via a relation to another entity so connected). In many ways, this can be thought of as the semantic reflex of syntactic dependencies and we can definitely expect UD to support this task.[1]

That said, it is well known that the basic UD representation does not consistently provide all the information that is needed to generate correct predicate-argument structures. Some of the deficiencies are remedied in the enhanced UD (E-UD), but there is a tradeoff with coverage, as only 31 out of the 213 UD treebanks contain useful E-UD edges.[2] This tradeoff becomes especially important in the context of universal semantic parsing (Reddy et al., 2017), i.e. an attempt to produce semantic representations (in our case, predicate-argument structures), in a universal way, relying only on the UD syntax and without using language-specific (e.g. lexical) resources.

In this paper, we try to assess how much E-UD helps with this task by asking to what extent it can be replaced with language-independent heuristics based on the basic UD alone.[3] The answer can inform practical decisions on how much effort to put into the creation of enhanced dependencies, and to guide future decisions on the development of the (E-)UD annotation. Because our goal is to support universal semantic parsing, we do not consider heuristics that rely on language-specific knowledge or resources.

[1] The relations should also be *labelled*, giving rise to a task of translating UD grammatical functions to appropriate semantic roles, but we do not consider this task here. Also, note that UD annotation does not allow us to identify eventualities introduced by non-verbal predicates (e.g. action nouns), so we ignore those in this paper.

[2] The TuDeT treebanks merely copy the basic dependencies over into the E-UD, while the Akkadian treebank only contains a single E-UD edge.

[3] There are many existing systems for augmenting basic UD dependency trees, and several whose effectiveness has been reported in the literature (Nyblom et al., 2013; Schuster and Manning, 2016; Nivre et al., 2018; Bouma et al., 2020). However, some of these are language specific, and several rely on machine learning. Here we report on the effectiveness of simple, algorithmic heuristics based on linguistic generalisations, and we apply them to a broad range of languages.

To evaluate our heuristics, we used the actual E-UD annotation as gold data and measured how well the heuristics reproduce this annotation from the basic UD. However, as it turned out, this approach is problematic because the E-UD annotations are often quite poor or inconsistent, both between and inside treebanks, making it hard to assess when the heuristic is wrong and when the annotation is wrong. Nevertheless, such cases still yield useful annotation recommendations.

The UD documentation specifies six types of enhancement:[4] empty (null) nodes for elided predicates, propagation of incoming dependencies to conjuncts, propagation of outgoing dependencies from conjuncts, additional subject relations for control and raising constructions, coreference in relative clause constructions, and modifier labels that contain the preposition or other case-marking information.

We will not deal with ellipsis in this paper, since its proper treatment is arguably semantic rather than (purely) syntactic (Dalrymple et al., 1991). The final type of enhancement, modifier labels, is entirely predictable from the basic UD graph and will not be further studied here either. Propagation of incoming dependencies is almost entirely predictable, except in cases of unlike function coordination, (Przepiórkowski and Patejuk, 2018). Worse, it is in fact not useful for semantic interpretation, but actually complicates matters, since it leads to the second conjunct having two incoming edges (`conj` and the copied edge), only one of which should be semantically interpreted.

This leaves three types of enhancement that we will deal with in the rest of this paper: control/raising (Section 2), propagation of outgoing dependencies (Section 3) and relative clauses (Section 4). For each of these types of annotation, we provide a theoretical discussion of how the E-UD can aid in obtaining a predicate-argument structure, quantify how well we can predict the E-UD from the basic dependencies with language-independent heuristics, and suggest changes to annotation policies and practices that would make E-UD more useful.

2 Additional subject relations for control and raising constructions

This enhancement adds dependencies which indicate the subject controllers of `xcomp`s. It is worth noting that the UD guidelines state that, to qualify as an `xcomp`, a predicate must participate in *obligatory control* (Williams, 1980), where its missing subject has to be interpreted as identical to a specified argument of another predicate (usually in a higher clause).[5] This is a fairly narrow understanding of control, and explicitly excludes cases of optional or arbitrary control (see Landau (2013) for explanation of these terms), which ought instead to be annotated as `ccomp`s or `advcl`s, according to context. We can see this as essentially restricting the UD annotation to the *grammatically determined* instances of control, in keeping with UD's role as a syntactic annotation scheme, and leaving processes such as anaphor resolution to the semantics.

There are two types of `xcomp` discussed in the guidelines: classic raising or control structures such as *I promised to come*, and secondary predications where the predicative component is a core argument of the main verb, such as *She declared the cake beautiful*. Figure 1 gives example annotations for these two structures, with the enhancement adding the controller indicated below the string:

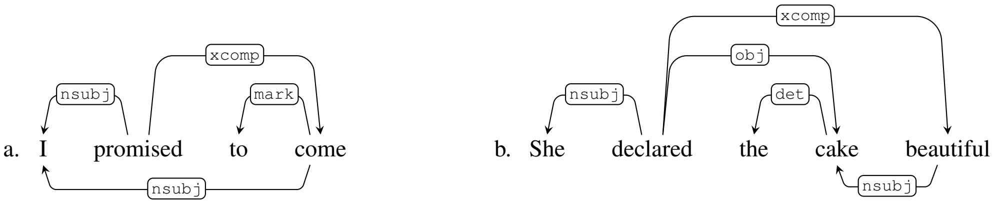


Figure 1: Examples of additional subject relations

In the basic annotation, there is no indication of the dependency between *come* and *I* (it is the speaker who will come), or between *beautiful* and *cake* (it is the cake which is declared to be beautiful); this is remedied in the enhanced representation.

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

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

Clearly, in order to obtain the correct predicate-argument structure for sentences like (0a) or (0b), these additional dependencies are necessary. And since these are not available in the basic UD tree, we do need the extra information added by the E-UD. What is more, the choice of external controller is, in principle, a lexical one – that is, it cannot be deterministically inferred from the syntactic structure of the sentence alone. For example, the a. and b. sentences in Figure 2 have identical basic UD trees (shown above the string), but the enhancements indicating the controller of the xcomp (shown below) differ, simply because the verbs in the main clause differ.

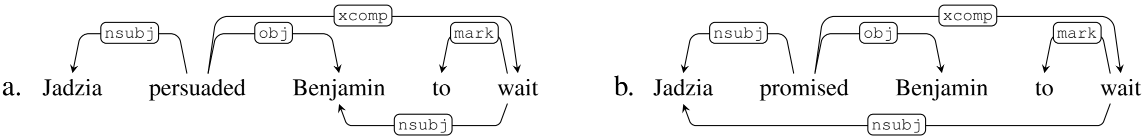

nsubj
xcomp
obj
mark
a. Jadzia persuaded Benjamin to wait
nsubj
nsubj
xcomp
obj
mark
b. Jadzia promised Benjamin to wait
nsubj

Figure 2: Lexically-determined control relations

Given this, an accurately annotated E-UD treebank would be particularly valuable for the purposes of semantic interpretation. However, only 22 of the UD treebanks contain this particular enhancement, and, as mentioned in the introduction, the quality of such annotations is not always high. So how successful can we be in adding such enhancements automatically to a basic UD annotation?

2.1 Heuristic

The linguistic generalisation we exploit here also appears in the UD guidelines' definition of xcomp: the xcomp's subject is controlled "normally by the object of the next higher clause, if there is one, or else by the subject of the next higher clause". That is, our heuristic assumes that if the head of the xcomp has an object dependent (obj, iobj, or ccomp), then that will be the controller;[6] if there is no object, then the subject, if present, will be the controller. If neither is present, we check whether the next highest head is itself an xcomp; if so, we continue to search upwards until we find a subject or object, or are no longer in an xcomp. This recursive search accounts for embedded xcomps, as in Figure 3:

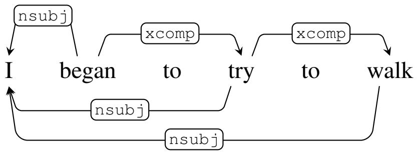


Figure 3: An embedded xcomp and its controller

Our heuristic is similar to the approach of Schuster and Manning (2016) and Nivre et al. (2018, 103), but with the addition of this recursive search in the case of embedded xcomps.

2.2 Results

Comparing the output of this heuristic against the E-UD annotations present in the 22 treebanks under discussion, we obtain an average precision score of 72.49% (see Table 1 for details). This is perhaps not terribly impressive. However, there are important caveats to consider. Firstly, the Dutch treebanks represent clear outliers (with 37.00% and 30.92%), and their removal increases the average by several percentage points (to 76.34%). The issues with Dutch appear to be because of systematic annotation errors in these two treebanks, where a number of xcomps do not have their controllers indicated, even though they are present in the string.[7]

[6] The relation ccomp is included as a kind of object here to account for examples like the following, from the English-GUM treebank, where the ccomp headed by *waking* is the csubj of the xcomp headed by *easier*: "It makes [waking up in the morning and getting out of bed at 6:00 a.m. when it's pitch black outside] [so much easier] when you're waking up really early" (GUM_vlog_london-18).

[7] Such problems are especially focussed around auxiliary-like predicates such as *lijken* 'seem', *liggen* 'lie'/'be', *blijven* 'remain', and *worden* 'become'/'be' (as passive auxiliary), whose xcomp complements very often do not have their controllers annotated. In many cases, it seemed to us that it might have been more more appropriate to annotate the complements of these verbs as the head of such constructions, and mark these verbs as aux instead, but we do not pursue this issue here.

Corpus name	Precision	Precision (controllers marked)	Recall
Albanian-TSA	66.67%	100.00%	66.67%
Belarusian-HSE	60.24%	76.70%	73.82%
Bulgarian-BTB	71.87%	98.49%	75.69%
Czech-CAC	67.00%	85.23%	78.12%
Czech-FicTree	63.03%	88.58%	78.83%
Czech-PDT	74.60%	87.89%	83.53%
Czech-PUD	55.36%	78.81%	75.61%
Dutch-Alpino	37.00%	92.97%	90.01%
Dutch-LassySmall	30.92%	94.57%	83.94%
English-EWT	93.84%	95.65%	90.76%
English-GUM	92.70%	99.46%	94.24%
English-GUMReddit	93.23%	99.20%	89.21%
English-PUD	92.00%	94.52%	88.09%
Finnish-TDT	56.89%	99.43%	60.29%
Italian-ISDT	76.94%	82.39%	80.28%
Latvian-LVTB	69.30%	95.03%	87.88%
Lithuanian-ALKSNIS	59.78%	93.79%	78.16%
Polish-LFG	95.26%	98.37%	94.41%
Slovak-SNK	68.81%	87.98%	84.03%
Swedish-PUD	87.62%	89.39%	84.29%
Swedish-Talbanken	86.31%	90.92%	86.13%
Ukrainian-IU	95.34%	98.50%	88.39%
AVERAGE	72.49%	92.18%	82.38%

Table 1: Performance of the heuristic used for adding external subjects

This points to a wider problem: in cases where the treebank in question contains an error, the precision score of the heuristic will suffer even when it is doing the right thing, linguistically speaking. There are cases where this occurs because of errors in the E-UD annotation, as with the Dutch examples, where controller annotations are omitted even when the controllers are present – here the heuristic often does a better job than the treebanks as annotated. There are also other cases where the quality of the *basic* UD annotations is the problem, and can mislead the heuristic: for example, where the treebank is right not to include a controller, but should not therefore have used an `xcomp` annotation in the first place. For instance, the following sentence shows an example of arbitrary control from the English-GUM treebank, which should have *chased* as a `ccomp` dependent of *what*, not an `xcomp`, as annotated: “Do you know what it’s like to be chased by the Ghost of Failure while staring through Victory’s door?” (`GUM_interview_messina-36`). If it were annotated in this way, the heuristic would (correctly) not look for a controller, and so would not (incorrectly) guess that it was the `nsubj` of *what*, namely *it*, and thereby hurt its precision score.

In both these cases, the only way to comprehensively determine to what extent the heuristic performs better than, or is unfairly misled by, the existing annotations would be through manual inspection. This is obviously time consuming, and also requires knowledge of many different languages, and so we have not been able to carry out such verification on a large scale. However, a sample analysis of 100 random errors from both the Dutch-Alpino and English-GUM treebanks indicates that our suspicions are borne out. Table 2 shows the sources of errors: overwhelmingly, the fault is with the annotation rather than with the heuristic. Most commonly this is because a controller is not annotated in the E-UD annotation when it should be (93/95 of the E-UD errors are of this kind in the Dutch corpus, 67/76 in the English). In all but one of these cases for each treebank, our heuristic correctly identifies the controller.

Corpus	Basic UD	E-UD	Heuristic	Not an error
Dutch-Alpino	3	95	1	2
English-GUM	25	74	1	1

Table 2: Sources of error in sample of 100 sentences from two corpora (numbers don’t sum to 100 because the 2 heuristic errors also involved E-UD errors)

Problems with the basic UD annotation constitute a sizeable minority in the English corpus, though they are rarer in the Dutch corpus. The majority of these are cases where the word which bears the `xcomp` dependency should have been annotated differently: either the construction in question involves non-obligatory control, so the dependency label should be `ccomp`;[8] or the dependent is a modifier not an argument (e.g. a purpose clause), so the label should be `advcl`;[9] or it is a secondary pred-

[8] As in the example mentioned earlier on this page.

[9] As in e.g. `GUM_news_asylum-7` from the English-GUM corpus: *Basya also believes the asylum seekers* [. . .] *may have left the boat on purpose to be* ***rescued*** *to* ***avoid*** *being sent away from Indonesia waters*, where putative `xcomps` are in boldface.

ication which is not a core dependent of the head, so it should be an acl.[10] The heuristic itself makes one error in each sample, and in both cases the E-UD annotation is also incorrect (because it does not include a controller at all).

Since the majority of errors we found in this sample analysis were due to the omission of controllers in the E-UD annotation, Table 1 also gives a precision score where the denominator is the number of guesses where the xcomp in question actually has a controller marked, rather than simply the total number of guesses, as a stand-in for a more thoroughgoing error analysis. Under these conditions, performance improves dramatically, to an average of 92.16% (and the Dutch outliers fall into line too). Of course, this may be concealing errors where the heuristic guesses a controller when one is genuinely not present; but such situations should be rare, given the definition of xcomp, and so we believe these figures are a fairer representation of the performance of this heuristic.

2.3 Discussion

There are of course cases where our heuristic will actually fail: with *promise*-type verbs as described at the start of this section, for example, since we assume that if the control verb has an object it must be the controller. Some languages might pose their own challenges too: for example, Nivre et al. (2018, 104) mention the fact that Italian allows (dative) obl controllers. This is not something we could easily incorporate into the heuristic as it stands, since the relation obl is used for verbal adjuncts as well as arguments. Judicious use of subtypes could help here, but we cannot guarantee that such subtypes would be present in a basic UD treebank.

We saw at the start of this section that the label xcomp is intended to be used for (grammatically governed) obligatory control. *Ceteris paribus*, we would therefore expect all xcomps to have controllers indicated in the E-UD annotations. However, as we saw above, this is not the case in the existing treebanks – the majority of 'errors' from our heuristic were cases where the heuristic identifies the correct controller but the treebank simply doesn't indicate one at all. One might therefore suggest that E-UD validation should include a check to ensure that all xcomps are properly controlled. However, this would not in fact be workable, because of the problem of implicit arguments. In some cases, the controller of an xcomp corresponds to an argument which is not realised in the string. This is especially pronounced in so-called *pro-drop* languages, where arguments of a verb (which ones will depend on the language) need not be realised overtly, with their referents being inferred either through context or through morphological marking on the verb itself. This is a problem for E-UD annotations of control because where an unexpressed argument of a higher predicate is the controller of an xcomp, it clearly cannot bear any relation to the xcomp in the enhanced representation (since it can bear no relations at all, not corresponding to a node in the string). In fact, this situation is not limited to pro-drop languages, and can also occur in a language like English, for example in a relative clause with no overt relative pronoun like *The man I told to leave* . . . , where the gap is the controller of the xcomp *leave*.

Having no controller marked on E-UD representations of certain xcomps is problematic for two reasons. Firstly, we lose linguistic information: we cannot capture the fact that control predicates enforce exactly the same kind of obligatory coreference between arguments when one of them is implicit as when it is explicit, because in the former case there is simply no node to be shared. There is no linguistic difference here at the relevant level of abstraction, but the annotation suggests there is. This representational divergence is undesirable from the point of view of UD's *universal* goals, and also makes downstream tasks such as semantic interpretation that much more difficult.

Secondly, the process of enhancing basic UD annotations, and of verifying that enhancement, is made more difficult. If the controller of an xcomp was always present in the string, the heuristic discussed above could be applied without missing implicit controllers. It would also be easier to verify enhancements or conduct error analysis, since any xcomp missing a subject edge in the E-UD could automatically be flagged as an error.

For these reasons, we agree with Patejuk and Przepiórkowski (2018, 216ff.) that including an 'empty' node in the basic UD representation for an implicit or gapped argument would be a major improvement

[10] As in e.g. cgn_exs\68 from the Dutch-Alpino corpus: *hij kwam* ***dronken*** *thuis* 'he came home drunk'.

to the expressivity and utility of the basic UD tree, and without adding too great a burden to the annotation task. This would ensure that all xcomps can have their controllers indicated in the enhanced UD annotations, thus harmonising the aforementioned differences across languages and constructions.[11]

3 Propagation of outgoing dependencies from conjuncts in coordinations

It is difficult to achieve a linguistically adequate annotation of coordination structures in dependency frameworks, as is widely accepted and extensively discussed in Popel et al. (2013). One particular problem is the distinction between modifiers that are private to one conjunct and those that apply to all, which is crucial to the creation of correct semantic representations: in *young capercaillies and grouses* we need to know whether *young* applies only to the capercaillies or also to the grouses, and in *shaved and brushed the cat* we need to know whether only the brushing or also the shaving applied to the cat.

If the conjunction is the head, as in some dependency annotation schemes, it provides an attachment point different from the individual conjuncts, and this can be used to make the annotation unambiguous.[12] But in schemes like UD where one of the conjuncts (the first, in the case of UD) is selected as the head, the problem becomes severe. In the basic annotation, shared dependents are attached to the first conjunct and so cannot in principle be distinguished from private dependents of the first conjunct. This underspecification is resolved in the enhanced dependencies, where dependents are attached to all the heads they belong to, so these are crucial for generating the correct predicate argument structure.

3.1 Heuristics

Disambiguation of shared dependencies potentially relies on very detailed contextual and encyclopedic knowledge. To correctly resolve the cases above, we need to know whether the context makes it likely that we are speaking about young grouses and whether it is normal to shave cats. This is way beyond the reach of language-independent heuristics. But in some cases we can make informed guesses.

First, valency information helps. UD does not directly express valency, but in situations where the potentially shared dependent bears a core or a functional relation (nsubj, obj, iobj, csubj, ccomp, xcomp, expl, aux, det, case, mark, cop) and the conjunct already has its own instantiation of that core dependent, we can be quite confident that the dependent is *not* shared. For the purposes of this heuristic, we can count nsubj and csubj as the same relation.[13] The resulting heuristic is purely negative, but can still be useful in restricting other, positive heuristics.

Second, because shared dependents are always attached to the first conjunct in the basic UD, the alternative, private dependent analysis looks quite different, depending on whether it is one where the dependent is private to the first or a later conjunct. Figure 4 shows the first case; the basic dependencies look the same for the private and the shared analysis, and we simply add an edge in the E-UD to express the shared conjunct. Figure 5 shows the second case, where the alternative analysis has the dependent being private to the second (or a later) conjunct. Here the alternative analyses differ also in the basic dependencies, because the dependent is attached to the first conjunct if it is shared, but otherwise to the second. This means that Figure 5a is unambiguous, and needs no enhancement.[14] The basic UD of Figure 5b, on the other hand, is ambiguous as long as we only consider the unordered tree, which is identical to that of Figure 4. But the word order disambiguates this case, since it shows that we only need to decide between a shared analysis and a second conjunct-only analysis, and the latter would look different already in the basic UD.

For English objects this is a foolproof heuristic: an object following the second verb cannot be private to the first conjunct. We speculate that this might also hold true in languages with freer word order, if

[11] Note that we are not proposing that the subjects of xcomps themselves be represented in the basic UD annotation, although this would also be possible. It would simply change the nature of the enhancement process: instead of adding a subject edge to the xcomp, we would have to connect the now already existing subject to its antecedent, perhaps with a ref dependency, as used in the E-UD analysis of relative clauses.

[12] This is done, for example, in the source treebank of Polish-PDB and Polish-PUD; see Wróblewska (2018).

[13] For languages like English, it would be tempting to treat expl as a subject relation, but this would hurt performance in treebanks where expl is used e.g. for “detransitivizing” object reflexive clitics; see Bouma et al. (2018).

[14] Notice that adding an object edge from *shaved* to *cat* in the E-UD would be an annotation error as the shared dependent should be attached to the first conjunct in the basic UD.

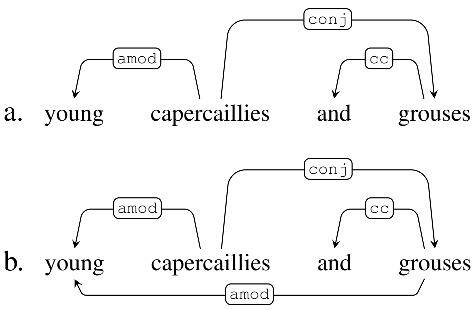

conj
amod
cc
a. young capercaillies and grouses
conj
amod
cc
b. young capercaillies and grouses
amod

Figure 4: Coordination where both analyses have identical basic UD trees

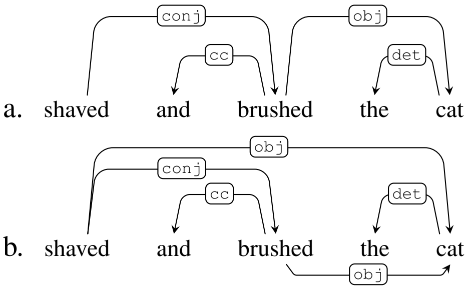


Figure 5: Coordination where each analysis has a different basic UD tree

nothing else because of a tendency to follow Behaghel's first "law": what goes together semantically goes together in the word order (Behaghel, 1932, 4–7). More generally, it is tempting to assume that if the potentially shared dependent belongs linearly to the second conjunct, but is annotated as a dependent of the the first conjunct, we are in the situation illustrated in Figure 5b; i.e. the dependent is shared and should be propagated. For the purposes of this heuristic, we take "belongs linearly to the second conjunct" to mean "occurs to the right of the leftmost word in the subgraph of the second conjunct".[15] Notice that we do not require that the dependent occurs to the right of the *head* of the second conjunct as it does in Figure 5b. If a language allows a word order like *shaved and the cat brushed*, *the cat* will count as belonging linearly in the second conjunct, and if it is annotated as a dependent of *shaved*, this will trigger a shared analysis. We will refer to dependents that are linearly in the second conjunct but are annotated as dependents of the first conjunct as *distant dependents*.

Finally, what if the potentially shared dependent belongs linearly to the first conjunct? In general, it is very hard to guess whether a dependent should be shared in this situation. Still, the strong universal tendency for verbs to always require a subject suggests that we can assume that the subject relations nsubj and csubj are always propagated to conjuncts that do not themselves have such a dependent. This may fail in cases where the second conjunct is an impersonal verb or in cases of pro-drop, as we will see in the error analysis.

In sum, this yields the following heuristic (**Heuristic 1**): never propagate a core dependent to a conjunct which has its own instantiation of that dependent, but otherwise **a)** always propagate subject relations and **b)** always propagate distant dependents. We will see that many treebanks have automatic enhancements which in many cases only propagate distant objects. For purposes of comparison, we therefore also include the results of a restricted version of Heuristic 1, which only propagates distant objects (**Heuristic 2**): never propagate a core dependent to a conjunct which has its own instantiation of that dependent, but otherwise always propagate subject relations and always propagate distant objects. This second heuristic has no linguistic motivation but merely aims to replicate automatic enhancements.

3.2 Results

Table 3 shows the performance of our two heuristics on the UD treebanks that have enhancements for propagation of outgoing dependencies.[16] For Heuristic 1, we report its overall performance, but also the performance of its two component parts.[17]

The treebanks clearly fall into two groups: for some treebanks (Bulgarian-BTB, English-EWT, English-PUD, Italian-ISDT, Swedish-PUD, Swedish-Talbanken) recall is over 95% for both heuristics and Heuristic 2 also achieves a precision of close to 100% (except in the case of Bulgarian-BTB). These are treebanks where the the propagation has been added by a heuristic very similar to our Heuristic 2 and, as such, the data are of little interest for assessing how well our heuristics can replicate gold stan-

[15] In the typical case, the leftmost word in the subgraph of the second conjunct will be the conjunction, which is a cc dependent of the head of the second conjunct, but other cases are possible e.g. when the conjunction is a clitic.

[16] We ignore Belarusian-HSE because there are only 25 scattered instances of propagation.

[17] Notice that there is some overlap between the components, as distant subjects will be propagated by both parts. Therefore, the recall of the whole heuristic will often be lower than the sum of the recalls of the parts.

	1: subj + dist		1a: all and only subj		1b: dist only		2: subj + dist obj	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Arabic-PADT	61.8	77.5	27.1	14.4	87.1	64.1	27.7	14.9
Bulgarian-BTB	40.2	100.0	63.6	100.0	0.2	0.2	62.2	100.0
Czech-CAC	81.7	50.0	64.9	18.5	95.3	33.0	66.7	20.2
Czech-FicTree	69.7	46.6	66.6	36.9	82.7	10.1	66.8	37.3
Czech-PDT	69.6	56.0	54.9	25.9	89.9	30.8	56.2	27.5
Dutch-Alpino	50.9	48.3	59.7	32.9	38.5	15.6	59.7	33.4
Dutch-LassySmall	50.8	48.3	51.6	32.7	49.7	16.0	51.6	32.7
English-EWT	61.1	100.0	98.7	93.4	10.9	7.7	98.1	99.1
English-GUM	59.6	83.7	97.9	78.2	10.0	6.1	97.6	83.3
English-GUMReddit	69.8	80.6	100.0	78.3	11.8	4.7	99.0	80.6
English-PUD	63.7	100.0	98.9	93.0	12.5	8.0	99.0	99.0
Finnish-TDT	84.5	38.9	84.3	26.1	85.2	13.2	84.8	27.2
Italian-ISDT	63.5	97.2	92.4	93.5	10.1	5.4	92.6	96.7
Latvian-LVTB	83.2	38.2	79.5	28.1	95.9	10.7	80.5	29.8
Lithuanian-ALKSNIS	59.9	36.1	48.3	19.3	77.4	18.2	51.7	22.1
Polish-LFG	69.5	31.9	67.9	29.6	100.0	2.9	68.2	30.0
Polish-PDB	81.0	34.9	72.8	21.4	98.6	13.8	74.1	22.8
Polish-PUD	87.9	33.4	81.8	20.7	100.0	13.4	82.7	22.0
Slovak-SNK	53.0	58.8	40.6	34.7	92.8	25.8	42.6	37.6
Swedish-PUD	63.6	100.0	100.0	93.8	11.3	7.3	100.0	100.0
Swedish-Talbanken	72.0	100.0	99.1	88.8	24.2	12.2	99.2	96.9
Ukrainian-IU	26.4	48.4	31.5	37.8	17.1	11.2	31.2	38.7

Table 3: Performance of propagation heuristics (bold-face = gold standard propagation enhancements)

dard annotation. English-GUM and English-GUMReddit also belong to this group, but the recall is lower because these treebanks also do some propagation of auxiliary verbs.

The other treebanks are more interesting. These are treebanks that arguably have genuine "gold standard" propagation of dependents. The Finnish and the Ukrainian treebanks have manually annotated enhanced graphs; but in the case of the Ukrainian treebank, the README reports that the annotation of propagated dependencies is only 40% complete, so we will disregard this treebank. The other treebanks have been converted from formats where shared dependencies were deterministically expressed (either Prague-style annotation, dependency schemes with the conjunction as the head, hybrid phrase structure/ dependency formats, or LFG). In principle, this was the case also with the Dutch treebanks, but here we discovered a number of conversion errors in the annotation.

For the other, gold standard treebanks, the precision of Heuristic 1 ranges from 53.0% on Slovak to 84.5% on Finnish, while recall ranges from 31.9% on Polish to 77.5% on Arabic. The propagation of distant dependents (1b) is a very sound heuristic in Polish (100% precision in two of the treebanks) and does quite well in Arabic, Czech, Finnish, Latvian, and Slovak (precision in the mid eighties or higher), but fares less well in Lithuanian (precision 77.4%). It naturally achieves very little recall on its own except in Arabic where it catches 64.1% of propagations. By contrast, subject propagation has a surprisingly low precision.

To understand better the behaviour of the heuristics, we performed manual error analysis of the 100 first precision errors[18] in the Lithuanian treebank, 50 errors in the propagation of subjects and 50 errors in the propagation of distant dependents. Table 4 shows the results.

As we see, annotation errors are by far the most common cause of precision errors by our heuristics. In 13 of 50 cases, the subject propagation rule adds a shared subject edge that should in fact have been there in the E-UD. In 19 cases, the error is in the basic UD leading to a misannotated structure, often involving a `csubj` that the heuristic propagates but which in fact should not be there at all. The actual linguistic errors are fewer (18), but of course more interesting. A characteristic of Lithuanian is the frequent use of impersonal verbs and those account for 12 cases where the heuristic propagates the subject of the first conjunct to the second conjunct which does not in fact take a subject at all. Many other kinds of subject shift can be detected with a simple feature check: the second verb is often in the first or second person.

[18] It makes little sense to explore the recall errors since we already know that the heuristics only cover a small proportion of possible shared conjunct structures.

But in this case both verbs are third person and so, for the basic UD to be unambiguous, we would need a feature `VerbType=Impersonal`.

	Impers. verb	Subj. shift	Basic UD	E-UD
1a) Subject	12	6	19	13
1b) Dist. dep.	—	—	48	2

Table 4: Sources of propagation errors in Lithuanian-ALKSNIS

For the propagation of distant dependents, the error analysis is more depressing in that all errors are in fact due to the annotation. It should be stressed that many of these involve "technical" relations such as `dep` and `flat` that are used in suprising ways. If we instead consider only the propagation of distant objects and obliques (106 cases), there is only a single precision error.

3.3 Insights

Word order turns out to be reasonably reliable as an indication of a shared dependency and most errors are due to misannotations. However, the coverage of this heuristic is quite limited. Subject propagation achieves much higher coverage, but its precision is low. Here too, the majority of errors are due to annotation errors. Some of these could be avoided with simple feature checking but, at least in the Lithuanian error sample, this would be much more useful if impersonal verbs were marked with a special feature.

4 Relative clauses

Relative clauses are clausal dependents of nouns, and hence bear the relation `acl` in the UD annotation. Semantically speaking, they represent unsaturated predicates, containing a gap which is either unrealised in the syntax or appears as a pronoun (relative or resumptive), which can either be *in situ* or displaced. They restrict the reference of the noun which heads the `acl` relation. For example, in interpreting the sentences *boys who Mary gave flowers* and *boys who gave Mary flowers*, we intersect the set of boys with the set of individuals that are respectively the goal or the agent of some giving event in the past.

The first condition for correct interpretation of relative clauses is therefore that we know there is a gap. The `acl` relation does not by itself provide this information, as it is also used for other clausal dependents of nouns (*a way to get my discount*, *the fact that nobody cares*). However the subtype `acl:relcl` is widely used in UD treebanks, and in this section we only consider this data.

4.1 Heuristic

Given that we know there is a gap, the next step in constructing the correct predicate-argument struture is the identification of the gap. This can sometimes, but not always, be done on the basis of the basic UD; by contrast, a proper E-UD annotation will always identify the gap. In fact, the E-UD representation is not so much an enhancement of the basic UD as a different theoretical perspective on relative clauses. The two analyses are shown in Figure 6:

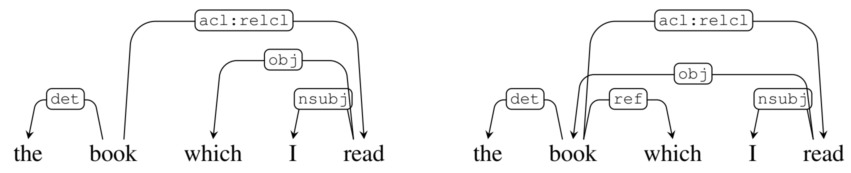

a. *Basic UD graph*   b. *Enhanced UD graph*

Figure 6: Two analyses of relative clauses

Figure 6a is what Falk (2010) calls a mediated analysis, i.e. one where the connection between the head of the relative clause and the gap inside the relative clause is mediated anaphorically by the relative pronoun. As a consequence, this connection is not represented directly in the syntax. By contrast, Figure 6b illustrates an unmediated analysis, where the head directly contracts a syntactic relation with the relative clause verb. Consequently, the graph contains a cycle, and the enhanced graph is not merely a straight augmentation of the basic graph. In addition, the relative pronoun becomes a `ref` dependent of the head. This is suggestive of the mediated analysis, but actually adds no information.

In the basic UD, the gap is only retrievable to the extent that the relative clause contains an identifiable relativizer, usually carrying the feature `PronType=Rel`. In such cases, it is straightforward to translate between the two analyses, and both would serve equally well as the basis for semantic interpretation.[19]

The case which distinguishes the approaches is the one where there is no relative pronoun; an example is shown in Figure 7. Here the E-UD graph has an argument dependency which is missing in the basic UD graph.

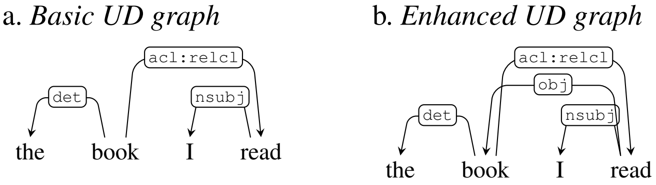


Figure 7: UD and E-UD analyses of a relative clause without a relative pronoun

The same problem may or may not arise in relative clauses introduced by a complementizer. For example, there is consensus in the grammatical literature that the word *that* in the variant *the book that I read* is a complementizer and not a pronoun filling the object position of *read* (Huddleston and Pullum, 2002, 1056f.). Nevertheless, the English treebanks consistenly treat it as a relative pronoun, allowing the relation of the gap to be expressed even in the basic dependencies. The Swedish treebanks do the same for *som*, while the related *som* in the Norwegian treebank is treated as a complementizer, thus prioritising giving the correct part of speech tag over expressing the gap that is needed for semantic interpretation in the basic UD (the Norwegian treebanks have no E-UD).

How can we guess the position of the gap if it is not present in the basic UD? Rule-based parsers typically use valency information to identify the missing argument, but this information is not present in the UD tree. We therefore rely on cross-linguistic tendencies as to what arguments are most accessible to relativisation, the so-called *Accessibility Hierarchy* (Keenan and Comrie, 1977), to determine which dependency is missing. The hierarchy is given in its original formulation in Figure 8a and translated to UD relations in Figure 8b.[20] Notice that we ignore genitives and objects of comparison, as they are rare and would not in any case be direct dependents of the `acl:relcl` verb, necessitating a further search.[21]

a. subject < direct object < indirect object < oblique < genitive < object of comparison
b. `subj` < `obj` < `iobj` < `obl`

Figure 8: The Accessibility Hierarchy and its translation into UD

The idea behind the heuristic, then, is to scan the dependents of the verb that bears the `acl:relcl` relation and assume that the gap bears the highest relation on the Keenan-Comrie hierarchy that is not present in the basic UD dependencies.

4.2 Results

Unfortunately, it turned out to be hard to evaluate this heuristic. First, few treebanks contain useful enhanced dependencies for relative clauses. 27 of the treebanks with E-UD do not have enhancements for relative clauses, or only have them when there is an overt relative pronoun, allowing them to be generated automatically but adding no new information. This leaves only 11 treebanks for our evaluation.

[19] Notice, however, that the E-UD can be interpreted directly from the graph, while the interpretation of the basic UD annotation relies on a lexical feature. Lexical features are less standardised than other parts of UD. We return to this point below.

[20] For the purposes of checking existing relations, we collapse `nsubj` and `csubj` into a single `subj` relation, since no predicate will have both. If this relation is missing, we assume it is `nsubj`, since relative clauses modifying clausal heads are rare, especially in the case of restrictive relative clauses, which is what we are considering here.

[21] We also ignore the possibility of ‘long-distance relativization’ as in *the book you asked Mary to look for*, which are also rare and necessitate a search for the correct attachment point.

Of these, Swedish-Talbanken, English-PUD, Italian-ISDT, English-EWT, Swedish-PUD and Estonian-EWT contain less than 10 instances of non-predictable enhanced dependencies for relative clauses. All these treebanks contain a large number of predictable enhanced dependencies and it seems that the scattered non-predictable E-UD edges are due to accidental omission of the feature `PronType=Rel` on the relative pronoun. Furthermore, the only non-predictable relative clauses in the Dutch treebanks are introduced by the relativizer *waar*, which bears the grammatical relation of the gap, but does not have the `PronType=Rel` feature. In such cases, the heuristic is doubly misled: first, it applies where it should not, because there is no `PronType=Rel` feature present in the clause, and next, it wrongly assumes that the grammatical function corresponding to the gap is actually filled. For example, in (8a), if *which* does not bear the the `PronType=Rel` feature, the heuristic will assume that we are in a pronoun-less relative clause and that the object position is filled, so that the gap is therefore `iobj`.[22]

Disregarding the treebanks where the only "informative" E-UD edges for relative clauses are due to accidental omission of the `PronType=Rel` feature, we have only three treebanks with non-trivial E-UD edges: Tamil-TTB, Ukrainian-IU and Belarusian-HSE. This indicates that the current E-UD annotation policy for relative clauses has not been very successful, as most treebanks either do not use it, or generate it only in the cases where it can be done automatically from the basic UD. We suspect the reason for this may be that it embodies a different theoretical perspective on relative clauses and therefore seems like an alternative analysis rather than a more informative one, even if it does it some cases contain more information.

Be that as it may, the results of applying our heuristic to the three treebanks that have non-trivial E-UD edges for relative clauses are shown in Table 5, and as we can see, they are decidedly mixed. We get good results on Tamil and Ukrainian and abysmal results on Belarusian. As it turns out, in 97.0% of the errors in Belarusian-HSE, the correct relation is `advmod`. The Keenan-Comrie hierarchy never predicts this, as it specifically addresses relativization on nominal positions.[23] But many languages use the equivalents of *where* and *when* to introduce clauses expressing location and time, and in many treebanks these are analysed as relative clauses. English is a case in point, but in the English treebanks these words are generally given the `PronType=Rel` feature (despite not being pronouns), hence making the gapped relation transparent. The Belarusian treebank, by contrast, does not add this feature.

	Belarusian-HSE	Tamil-TTB	Ukrainian-IU
success	1	368	65
failure	202	28	4

Table 5: Evalutation of heuristic for relative clauses

4.3 Discussion

The E-UD representation is crucial for a correct semantic analysis of relative clauses where the gap cannot be identified by the `PronType` feature. However, very few treebanks contain such enhanced dependencies; in practice, the enhanced dependencies are only generated when they can be unambiguously derived from the basic UD. This suggests that here too a limited use of empty nodes could be beneficial in allowing for the expression of the gap in the basic UD, even when there is no overt relative pronoun. Figure 9 shows what the annotation would look like.

This would make it possible to consistently give an interpretable annotation of relative clauses in the basic UD, and render the enhanced version superfluous.

[22] Another problem is that many treebanks use the the multivalue feature `PronType=Int,Rel` (reflecting the interrogative/ relative ambiguity that is common in Indo-European and beyond), although the UD guidelines specify that these should be used sparingly and only when one cannot decide between the two features. If the clause itself is marked as `acl:relcl`, it is of course clear that the *wh*-word that introduces it is a relativizer and not an interrogative, so `PronType=Rel` should have been used. However, precisely for that reason, it seems safe to interpret `PronType=Int,Rel` as indicating a relative pronoun in this context. Our experience suggests that it is even safe to interpret the wrong tag `PronType=Int` as meaning `PronType=Rel` inside an `acl:relcl` subtree.

[23] Note that even if we added `advmod` to the bottom of our prediction hierarchy, our heuristic would only add it to verbs that already have `subj`, `obj`, `iobj` and `obl` dependents.

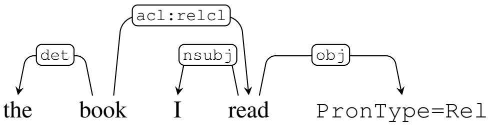


Figure 9: Null relative pronoun annotated in basic UD

5 Conclusion

Overall, it is clear that some enhancements of basic UD annotations are necessary in order to derive correct predicate-argument structures. E-UD does offer these, but there are two important limitations. The first is coverage: only 31 treebanks have any (useful) E-UD annotations, and even fewer contain all six subtypes identified by the UD guidelines. The small size of this selection is further compounded by it being more typologically restricted than the impressively global spread of UD: those treebanks with E-UD are much more European and much less diverse (of the 31 treebanks, 4 are English, 4 are Czech, 3 are Polish, . . .). The second limitation is quality. In our investigations, we found that the E-UD annotations were inconsistent at best, and often the result of limited automatic processes with minimal manual verification – although we have only quantified these shortcomings in a very preliminary way.

One solution to these limitations would be to invest time and resources into improving the quality of existing E-UD annotations. For the propagation of outgoing dependents in coordinations, this may in fact be the only feasible solution. For control and relative clauses, however, we suggest another approach. As we noted above, the addition of empty nodes for certain phenomena in the basic UD annotation would allow for the automatic generation of an improved E-UD annotation, or even make it redundant, since the basic UD would now contain the missing information already. Noting that the existence of empty nodes has already been sanctioned in the E-UD treatment of ellipsis, we suggest that generalising this to other phenomena in the basic UD annotation could be much more worthwhile than annotating enhanced edges independently.

Acknowledgements

This work was supported by the Research Council of Norway, grant number 300495 "Universal Natural Language Understanding". We are grateful to the three reviewers for their feedback on the previous version and to Alicia Cleary-Venables for help with the Dutch data.

References

Otto Behaghel. 1932. *Deutsche Syntax IV*. Carl Winter, Heidelberg.

Gosse Bouma, Jan Hajic, Dag Haug, Joakim Nivre, Per Erik Solberg, and Lilja Øvrelid. 2018. Expletives in Universal Dependency treebanks. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 18–26, Brussels, Belgium, November. Association for Computational Linguistics.

Gosse Bouma, Yuji Matsumoto, Stephan Oepen, Kenji Sagae, Djamé Seddah, Weiwei Sun, Anders Søgaard, Reut Tsarfaty, and Dan Zeman, editors. 2020. *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, Online, July. Association for Computational Linguistics.

Mary Dalrymple, Stuart M. Shieber, and Fernando C. N. Pereira. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14(4):399–452.

Yehuda N. Falk. 2010. An unmediated analysis of relative clauses. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG10 Conference*, pages 207–227. CSLI Publications.

Irene Heim and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Number 13 in Blackwell Textbooks in Linguistics. Blackwell, Oxford.

Rodney D. Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press, April.

Edward L. Keenan and Bernard Comrie. 1977. Noun phrase accessibility and Universal Grammar. *Linguistic Inquiry*, 8(1):63–99.

Idan Landau. 2013. *Control in generative grammar: a research companion*. Cambridge University Press, Cambridge, GB.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Joakim Nivre, Paola Marongiu, Filip Ginter, Jenna Kanerva, Simonetta Montemagni, Sebastian Schuster, and Maria Simi. 2018. Enhancing Universal Dependency treebanks: A case study. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 102–107, Brussels, Belgium, November. Association for Computational Linguistics.

Jenna Nyblom, Samuel Kohonen, Katri Haverinen, Tapio Salakoski, and Filip Ginter. 2013. Predicting conjunct propagation and other extended Stanford dependencies. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 252–261, Prague, Czech Republic, August. Charles University in Prague, Matfyzpress, Prague, Czech Republic.

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

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

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.

Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 89–101, Copenhagen, Denmark, September. Association folr Computational Linguistics.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2371–2378, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Edwin Williams. 1980. Predication. *Linguistic Inquiry*, 11(1):203–238.

Alina Wróblewska. 2018. Extended and enhanced polish dependency bank in universal dependencies format. In Marie-Catherine de Marneffe, Teresa Lynn, and Sebastian Schuster, editors, *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 173–182. Association for Computational Linguistics.

Starting a new treebank? Go SUD!
Theoretical and practical benefits of the Surface-Syntactic distributional approach

Kim Gerdes
Lisn, CNRS,
Université Paris-Saclay
gerdes@lisn.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

The paper brings to the fore some advantages to first develop a new treebank in Surface-Syntactic Universal Dependencies (SUD) annotation scheme, even if the goal is to obtain a UD treebank. Theoretical benefits of SUD are presented, as well as UD-compatible SUD innovations. The two-way UD ⇔ SUD conversion is explained, as well as the possibility to customize the conversion for a given language. The paper concludes by a practical guide for the development of a SUD treebank.

1 Introduction

SUD, Surface-Syntactic Universal Dependencies, is a syntactic annotation scheme, which is a convertible variant of Universal Dependencies (UD). UD is a very successful treebank development project that is now an indispensable standard of data-based syntax (de Marneffe et al., 2021). To benefit from UD's wealth of expertise, tools, and cross-language comparability, any annotation scheme must eventually be convertible into UD. Nevertheless, the UD annotation scheme was initially developed in the context of NLP applications, rather than pure linguistic considerations and some initial choices are problematic.[1] SUD is based on a different theoretical framework that has many advantages for treebank development as we will show in this paper.

SUD has already been presented in two papers by (Gerdes et al., 2018; Gerdes et al., 2019). While SUD's theoretical foundations remain unchanged, this paper proposes one change of SUD's philosophy. At first, SUD was thought of as a pure variant of UD with a complete equivalence between SUD and UD. Initially, SUD was more interested in the UD ⇒ SUD conversion because for some studies, especially on word order typology, a more surfacic annotation was required.[2] This paper reports on a growing interest in SUD ⇒ UD conversions and the development of treebanks in SUD in order to obtain both SUD and UD variants of the treebank. The UD ⇒ SUD conversion grammar is still maintained and has even been improved with the possibility to more easily customize the conversion for a given language. Recent views on SUD abandons the idea of having an equivalence between the two annotation schemes, and this

[1] UD is initially based on Stanford dependencies, which was itself the conversion into a dependency tree of the outputs of a phrase-structure-based parser. In consequence, UD dependency relations combine both functional and categorical information, for instance with the nsubj vs csubj distinction between nominal and clausal subjects, the obj vs ccomp distinction between nominal and clausal objects, or the amod vs nmod vs advmod distinction between adjectival, nominal, and adverbial modifiers, as well as the obl vs nmod distinction between adpositional phrases depending on a verb or a noun. Moreover, UD is very semantically-oriented, favoring relations between content words, leaning towards a sort of interlingua representation. The part of speech tags, stemming from Google's universal POS (Petrov et al., 2012) and the Interset interlingua tagset (Zeman, 2008), were added independently, resulting in some redundancy.

[2] Let us recall that in UD function words depend on content words. As a consequence, adpositions are dependents of the noun with which they form a phrase. This is in complete contradiction with typological studies that show that the adposition-noun relation tends to have similar properties than the verb-object relation. In particular, VO languages have prepositions while OV languages have postpositions (Dryer, 1992).

paper postulate that SUD is a richer annotation scheme than UD. In other words, no information is lost in UD ⇒ SUD and a double conversion UD ⇒ SUD ⇒ UD should give the initial treebank, eventually with additional features.[3] But a SUD ⇒ UD conversion generally causes a loss of information and SUD treebanks obtained from a UD conversion are underspecified for some features considered as relevant for the SUD annotation scheme, such as the internal structure of nuclei or of MWEs. As a simple example consider the verbal chain in the sentence *I would have left*. SUD annotates the hierarchical relation between the three verbs (*would* → *have* → *left*), UD sees a flat structure in these three verbs with the lexical verb (*left*) at its head. Therefore, the hierarchical relation between *would* and *have* is not encoded in UD, and requires language specific heuristics to obtain the correct SUD structure. Theoretical benefits of SUD are presented in Section 2 and completed in Section 3 by UD-compatible SUD innovations.

Due to the fact the SUD is richer than UD, we encourage developers of treebank to start with a SUD annotation, which allows them to obtain a high-quality UD treebank, while keeping information that is flattened out in UD. Moreover if a treebank already exists in a third format, it can be easier to convert it into SUD and only then into UD rather than to aim UD directly because of the unconventional lexical-word-centric approach of UD. We may further assume that SUD's additional richness does not slow down the overall annotation process as it also removes some redundancies of UD. The UD ⇒ SUD and SUD ⇒ UD conversions are presented in Sections 4 and 5, as well as the possibility to customize the conversions for a given language. Section 6 sketches a practical guide for the development of a SUD treebank.

2 Theoretical benefit of SUD

We discuss four benefits of SUD compared to UD: a definition of dependency based on distributional criteria, an encoding of the internal structure of nuclei, a definition of syntactic relations based on commutation positional paradigms, and a more symmetrical analysis of coordination. These properties are core elements that cannot be integrated in UD, which is based on different fundamentals. Other benefits of the current SUD annotation that could be adopted in UD are presented in Section 3.

2.1 Definition of dependency based on distributional criteria

UD favors relations between content words, while function words are treated as dependents of content words. While it may seem at first view that it is easy to establish the difference between function and lexical words for a new language, it turns out to be a hard task to delimit the content word - function word opposition that is compatible with a coherent non-catastrophic annotation.[4] Moreover, supposing that the opposition is semantic or language independent can lead to erasing typologically important structural differences, for example when languages differ precisely in the structure of function words. Relegating all function words as done by UD makes us loose some syntactic information as we will see in the next section.

SUD favors a definition of the dependency structure based on a more traditional definition of head: The head of a unit U is the element A that controls the distribution of U. By *distribution*, we mean what Mel'čuk's (1988) calls the *passive valency*, that is, the set of possible syntactic governors for U, or, similarly, the set of syntactic positions that U can occupy. Even if the notion of governor is based on the notion of distribution, we avoid the circularity, because in most cases the question of the head is not controversial, especially for the governor of a sentence.

As soon as we can determine units and a head for each unit, we have a dependency structure (Gerdes and Kahane, 2013): B depends on A as soon as A is the head of the unit that A and B form together.

This definition of the head is based on formal criteria that we want to recall here because they have often been misstated. Let us consider a unit U = AB. The simplest case is when A or B can stand alone.

[3] The lossless conversion might require language-specific rules, see Section 4.

[4] We use *catastrophe* here in a strictly mathematical sense of Thom's catastrophe theory (Saunders, 1980), i.e. a brutal structural change in a continuum. In the case of annotation, this boils down to very similar constructions ending up with very different syntactic structures, see (Gerdes and Kahane, 2016) for details.

In this case the distribution of A or B can be considered and compared with the whole unit U.[5] It gives us two criteria.

Positive distributional criterion with deletion. If U = AB, A can stand alone (i.e., B can be deleted), and U and A have the same distribution, then A is a head of U.

Negative distributional criterion with deletion. If U = AB, B can stand alone, and U and B do not have the same distribution, then A is a head of U.

The second criteria can be applied to examples such as U = *John ran* or U' = *with John*, where B = *John*. Clearly B does not have the same distribution as the clause U or the phrase U' and then the verb is the head of U and the adposition the head of U'. In the same way, a combination auxiliary-verb such as U = *is expected* has the auxiliary as head, because the past participle has a different distribution: It can be the dependent of a noun (*that's the guy expected at noon*), while *is expected* can be the dependent of a verb (*he knows he is expected*).[6]

It is not needed to delete an element to decide which element is the head, a commutation with another element is sufficient:

Distributional criterion without deletion. If U = AB, A can commute with an A', and U and U' = A'B does not have the same distribution, then A is a head of U.[7] In other words, if B depends on A, then B must not modify the distribution of A and a commutation on B does not change the distribution of the unit it forms with A.

For instance, U = *with John* and U' = *by John* have different distributions. In other words, the commutation of *with* and *by* change the distribution, which implies that the preposition is the head. The same criteria can be used with the determiner-noun combination: Some nouns such as *day* (*she stayed two days*) or *time* (*I will do that (the) next time*) have a very special distribution, being able to work as an adverbial phrase, whatever the determiner is. This is a good argument to take the noun as the head, even if there are also arguments to take the determiner as the head.

2.2 Internal structure of nuclei

In a recent paper on UD, de Marneffe et al. (2021) justify treating function words as dependents as follows: "Sometimes linguistic head functions are divided between a structural center (an auxiliary or function word) and a semantic center (a lexical or content word), such as for periphrastic verb tenses like *has arrived*. This is what Tesnière (2015 [1959], ch. 23) refers to as a dissociated nucleus. In such cases, UD chooses the lexical or content word as the head, and makes function words dependents of the head in the dependency tree structure, while recognizing that they do form a nucleus together with the content word." Nevertheless in case of the presence of multiple function words, Tesnière considers that there is an embedding of nuclei, while UD only considers a flat structure with all function words depending on the same content word and the internal structure of the nucleus is completely lost. For instance, in the sentence of Fig. 1, *in Mesoamerica* is clearly a nucleus that is put in a comparison with *in the Americas* and then embedded in *than in Mesoamerica*.[8] The UD analysis does not have a phrase *in Mesoamerica*.

In the UD ⇒ SUD conversion, we use heuristics that are described in Section 4, depending on the order of the function words and their function. In particular, the closer a function word is to the content word, the earlier they combine. The SUD structure of the same sentence is given in Fig. 1 (lower part).

[5] When comparing the distribution of two units, we mainly use our intuition. For tricky cases, we also observe the actual distribution in our corpora, but nothing is completely currently formalized.

[6] *is expected* can also be the dependent of a noun, but only if it combine with a relativizer (*the guy that is expected at noon*) and, in this case, it is the relativizer that is head of the relative clause, because the relativizer change the distribution of *is expected*.

[7] When saying that A' can commute with A, we are only considering the commutation in the context of B. In other words, this means that A'B is a valid combination and that A and A' exclude each other in this context (i.e. AA'B is not valid).

[8] Note that the analysis of comparative complements is erroneous in English UD treebanks: *than in Mesoamerica* should depend on *more* and not on *obvious*, because *more than in Mesoamerica* is a valid sub-unit of the sentence and not **obvious than in Mesoamerica*.

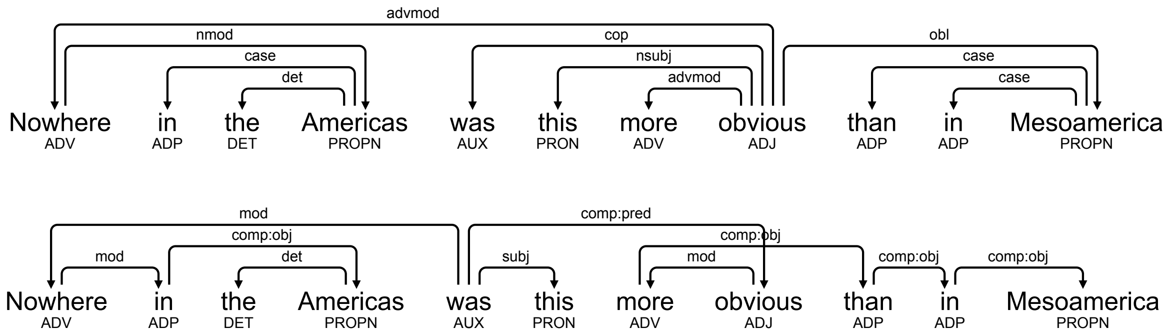


Figure 1: UD and SUD analysis of Sentence *Nowhere in the Americas was this more obvious than in Mesoamerica.* (GUM_textbook_history-19)

But this heuristic does not work in some cases. For instance, Wolof has a multitude of auxiliaries that are used to focus the subject, a complement, or the verb itself, which will occupy the first place in the clause (Robert, 1991; Bondéelle and Kahane, 2021). The auxiliary *na*, used to focus a verb, can also focus an auxiliary, as in Fig. 2 where the past imperfective auxiliary *doon* is focalized by *na*, which is the head of the nucleus *doon na* VERB. Here, *na* is the closest function word to the content word, but it combines last.

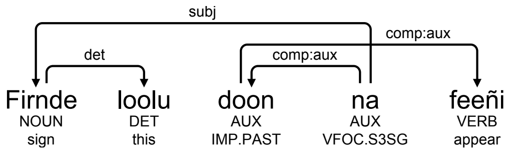


Figure 2: SUD analysis of the Wolof sentence *Firnde loolu doon na feeñi* ‘This sign was to be revealed.’

Another problematic case is when there are function words on both sides of the content word. This can be illustrated by the auxiliaries in German, as in sentence (1).

(1) *Jeder siebte Beschäftigte wird dann seine Kündigung erhalten haben*
Each seventh employee will then his notice received have
‘One in seven employees will have received their notice by then.’

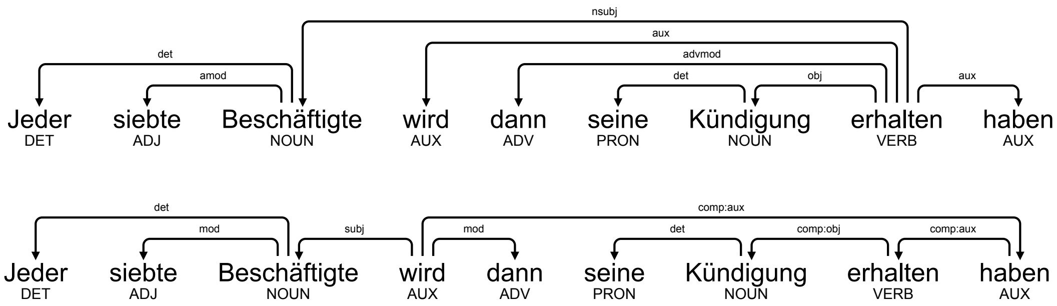


Figure 3: UD and SUD analysis of Sentence (1)

German is a V2 language, where the finite verbal form always occupies the second position of a declarative sentence, whether it is a content verb or an auxiliary. In (1), the verb has two auxiliaries, *wird* ‘will’ on the left and *haben* ‘have’ on the right. The auxiliary on the left, which is in the second position in the sentence and has a finite form, is the root of the syntactic structure, which cannot be guessed from the flat UD structure alone.

2.3 Definition of syntactic relation based on positional paradigms

In SUD, two dependents that belong to the same positional paradigm have the same syntactic relations, in accordance with Mel'čuk's (1988) or Van den Eynde & Mertens' (2003) definitions, while UD takes also into account the POS of the governor and/or the dependent (see Note 1 about the definition of relations in UD). One advantage of the SUD definition is the possibility to compare the valency of two occurrences of the same lemma and to extract a syntactic lexicon more easily.

As UD, SUD uses the notation `rel:subrel` for a sub-relation of a given relation. Syntactic relations are part of a hierarchy and `comp:obj` or `comp:obl` must be understood as sub-relations of a more generic `comp` relation. Modifiers (`mod`) and complements (`comp`) are distinguished, but a super-relation `udep` (underspecified dependency) can be used if we do not want to make this distinction. We use it for noun dependents and it is used in non-native SUD treebanks for the conversion of the UD `obl` relation, which gives the `udep` relation in SUD.[9] Figures 4, 5, and 6 give UD and SUD annotations of verb dependents which are respectively modifier, argument and underspecified. Annotations in Figures 4 and 5 are SUD-native and contain a distinction between complements and modifiers, which is kept in the conversion with the UD relations `obl:arg`, `iobj`, and `obl:mod`. Conversely, the sentence in Figure 6 comes from UD_English-GUM, where the distinction between complements and modifiers is not present for preposition phrases and the conversion to SUD gives us a `udep` relation.

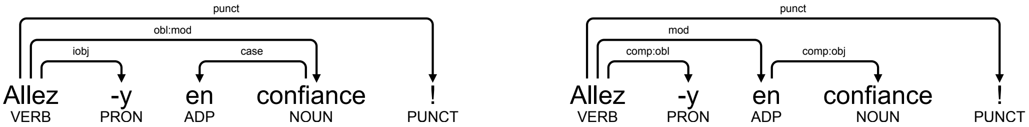


Figure 4: UD and SUD analysis of *Allez-y en confiance !* 'Go there with confidence!'

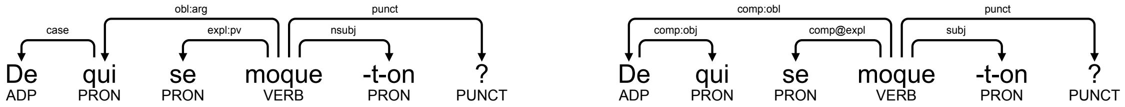


Figure 5: UD and SUD analysis of *De qui se moque-t-on ?* 'Who are we kidding?'

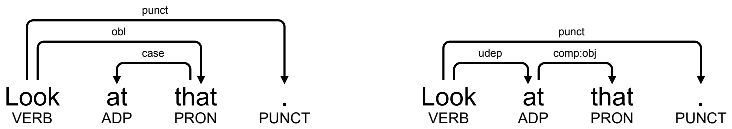


Figure 6: UD and SUD analysis of *Look at that.*

Additional features on relations are clearly separated from the relation itself, especially when it is semantic information. We use for this the delimiter `@`. For instance, the semantic value of an auxiliary (tense, passive, causative) can be indicated on the `comp:aux` relation: `comp:aux@tense`, `comp:aux@pass`, `comp:aux@caus`. Subjects all have the function `subj`, but expletive or passive subjects can be marked by an additional feature: `subj@expl`, `subj@pass`.[10] In spoken corpora, the feature `@scrap` has been used for incomplete units. This feature is particularly useful for error mining:

[9] UD uses the `obl` relation for all adpositional phrases depending on a verb, but for clauses depending on a verb, a distinction is made between complements (`ccomp` or `xcomp`) and modifiers (`advcl` for adverbial clauses).

[10] Contrary to UD, SUD does not have an `expl` relation for expletives. We consider that *it* in *it is impossible to do that*, is above all a normal subject and is analysed as `subj@expl`.

for instance, a relation between a verb and a determiner (in an incomplete sentence such as *I see the…*) should not be allowed without a @scrap.

2.4 A more symmetrical analysis of coordination

In UD, the dependent shared by all the conjuncts are attached to the head of the coordination, the leftmost conjunct.

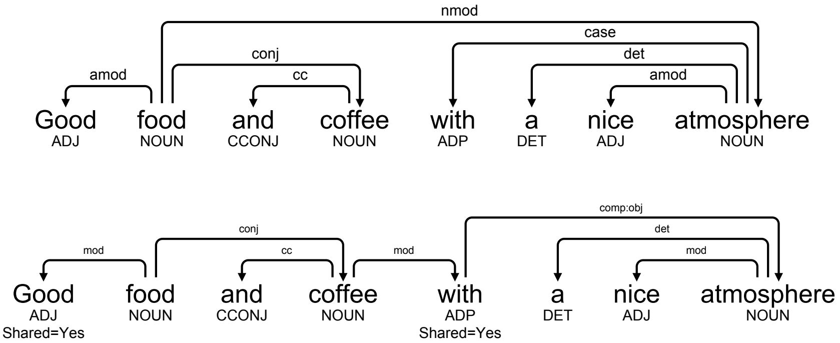


Figure 7: UD and SUD annotation of *Good food and coffee with a nice atmosphere*

In the example of Fig. 7, from the UD_English-EWT corpus, there are two modifiers of the coordination *food and coffee*: a left modifier *Good* and a right modifier *with a nice atmosphere*. Since the right modifier is after the second conjunct, the UD annotation has only one interpretation: It cannot be the modifier of the first conjunct alone but only of the coordination as a whole. However, for the left modifier, the UD annotation does not indicate whether it is a modifier of *food* only or of *food and coffee*. This is an unfortunate asymmetry.

In SUD, as in UD, the head of the coordination is the head of the leftmost conjunct, but for the dependents, the annotation is perfectly symmetrical. They are attached to the nearest conjunct: the left to the leftmost conjunct and the right to the rightmost conjunct. In order to indicate which dependents are shared, we introduce the feature Shared with values Yes and No. Conversions of UD treebanks, only give a partial instantiation of the Shared feature. In the native SUD_French-GSD, Shared=Yes features have been systematically introduced. Note also the considerably shorter overall dependency lengths of the SUD annotation scheme, which is not only cognitively more plausible but also facilitates manual annotation and correction.

3 UD-compatible SUD innovations

This section presents features of the SUD annotation scheme that could, and we believe should, be integrated into the UD annotation guidelines. For now, the SUD ⇒ UD conversion will encode these SUD features as optional additional information in the MISC column.

3.1 Internal structure of Multi-Word Expressions

Multi-Word expressions (MWE) cover a wide heterogeneous field of constructions such as use of foreign words that have no internal structure in the host language (*Burkina Faso*, *Hong Kong*, *ad hoc*), or completely regular structures in named entities (*the Embassy of Ecuador in London*, *the United States*). Interesting from a syntactic point of view is another set of phenomena: constructions that have a regular internal structure but that intervene as a whole at an unexpected point in the sentence. For example, *in order* (*to* VERB) is analyzed as an MWE in English treebanks, as shown in Fig. 8 (upper part) from UD_English-GUM, with a fixed relation between *in* and *order*.

Even if *in order* is semantically frozen it is nevertheless a syntactically regular preposition-noun combination. In native SUD, the sentence is analyzed with the standard comp:obj relation between *in* and *order* (the noun is the object of the adposition) and the idiomaticity is encoded by additional features Idiom=Yes on the head and InIdiom=Yes on the other elements.

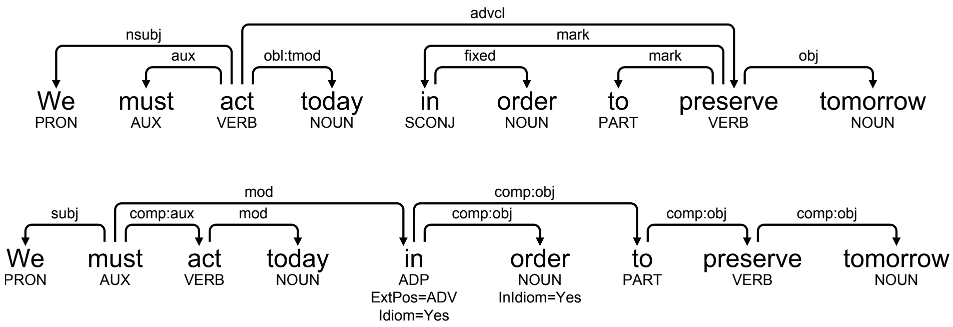


Figure 8: UD and SUD annotation of Multi-Word Expressions

Moreover, we consider that *in order* as a whole works as an adverb, which is encoded in SUD by the feature `ExtPos=ADV` (for external POS).[11] Of course, this SUD analysis translates into a different UD analysis, because adverbs are analyzed as content words.[12] Arguably, the UD analysis would have been different if the internal structure of the MWE had been taken into account.

3.2 Textform and wordform

It was identified in UD that, in several places, syntactic units do not exactly correspond to orthographic units given in the raw text.[13] For instance, in French the orthographic unit *au* is a contraction of two syntactic units: the preposition *à* and the determiner *le* (such amalgams are called Multi-Word Tokens or MWT). With a focus on syntax, it is natural to consider syntactic units as the basic units of annotation; this is what is done both in UD and in SUD. However, it is necessary to keep all the information and to also encode the orthographic unit when it differs from the units of the structure. The UD guidelines[14] introduce the CoNLL-U format with a dedicated mechanism with a new type of line describing a range of tokens (`2-3` in the example below) to store the contracted form.

```
2-3 au _ _
2 à à ADP
3 le le DET
```

The main drawback of this solution is that the syntactic dependency structure, being based on the syntactic units, does not refer to the orthographic units which are then not easily accessible for tools working on the syntactic structure. Having access to these orthographic units is useful for parsing.

There are other cases where an orthographic unit is different from a canonical token. For instance, for several languages, uppercase letters are used at the beginning of a sentence, in specific usages for naming institutions (*the White House*), in titles (*What the Moon Brings* [GUM_fiction_moon-1]), or for emphasis (*YES!*). It is useful to encode the canonical form in these cases, as it allows for an improved data analysis, performing linguistic queries on canonical forms.[15]

We propose a new way to encode the orthographic information in these two cases (MWTs and non-canonical forms) with two new features: `textform`, which always contains orthographic data and `wordform` which always contains a canonical lexical form (see Table 1 for examples).

[11] English has adverbs taking a *to* VERB complement, such as *up*, *next*, *about*, or *prior*, but there are no subordinating conjunctions with this valency.

[12] In order to keep a function word status for *in order*, *in* has been analyzed in the UD analysis of Fig. 8 as a subordinating conjunction (`SCONJ`, as in all occurrences of *in order* in UD_ENGLISH-GUM, version 2.8), which is surprising to say the least.

[13] Here, *orthographic* means the actually observed letters in input text.

[14] `https://universaldependencies.org/format.html#words-tokens-and-empty-nodes`

[15] Note that this canonical form may not be trivial to recover. In French, diacritics are optional on upper-case letters, and an *A* as the first word can be either the preposition *à* (ex: *à qui tu penses ?* 'who are you thinking of?') or a verbal form *a* (*a-t-il choisi ?* 'has he chosen?').

	`form`	`lemma`	`textform`	`wordform`	`CorrectForm`
[fr] au	à	à	au	[à]	
	le	le	_	[le]	
[en] wanna	want	want	wanna	[want]	
	to	to	_	[to]	
[en] The	The	the	[The]	the	
[fr] Le maison	Le	le	[Le]	le	La
[en] egg plant	egg	egg	[egg]	eggplant	
	plant	plant	[plant]	_	
[en] NEEEVERR	NEEEVERR	never	[NEEEVERR]	neeeverr	never

Table 1: Examples on the usage of features `textform`, `wordform` and `CorrectForm`.

The main advantage is that, using features, all information is available in the units used in the syntactic structure and it makes it possible to use these features in any tool (for querying the treebank, for conversion. . .).

It might seem appealing to use these features for encoding typos as well. But, there may be conflicts, as shown for the phrase [fr] *Le maison*: *Le* must be corrected in *La* (the gender of *maison* is feminine) but also be normalised into *le*. So, we decided to use the feature `CorrectForm` (already used in other UD treebanks) in case of typos, to express the way it should be written.

In order to avoid having an overly verbose CoNLL file, we propose in practice, to explicitly record `textform` and `wordform` only when they are different from the feature `form` (column 2 in CoNLL). In Table 1, square brackets are used to show feature values which are not stored in the CoNLL file.

4 The conversion UD ⇒ SUD

Our approach of the conversion between different syntactic annotations is based on graph rewriting. Each annotation is seen as a graph and the conversion of an annotation into another annotation is performed by applying a sequence of local graph rewriting rules. For this, we use the GREW tool[16]. In Grew, a Grew Rewriting System (GRS) is a set of rewriting rules organized into strategies such that these rules can be ordered, iterated and grouped into packages.[17]

Since SUD is richer than UD, a universal UD ⇒ SUD GRS can only approximate the correct SUD annotation due to the lack of information in the UD annotation, and the adaptation of the GRS to each language is crucial.

4.1 The universal conversion UD ⇒ SUD

The universal UD ⇒ SUD system has five main tasks to perform:

1. Replacing UD dependency labels with SUD dependency labels.
2. Reversing some dependencies between function words and lexical words to change the heads of adpositional phrases, subordinate clauses, and verb-auxiliary pairs.
3. Shifting the source of some dependencies as the result of reversing dependencies.
4. Attaching the right dependents of coordinations to the rightmost conjunct, whereas in UD they are attached to the leftmost conjunct, the coordination head (see Section 2.4).
5. Transforming bouquets of coordinated elements into sequences, marking embedded coordinations with the `emb` extension added to `conj` relations.

[16] `https://grew.fr/`

[17] All GRS described in this section are available on `https://github.com/surfacesyntacticud/tools/tree/master/converter`

These tasks are not independent of each other and although they can most often be carried out in any order, their forms depend on this order and sometimes one order is more relevant than another. The universal UD ⇒ SUD GRS contains 89 rules grouped into 20 packages.

As said above, a conversion of an UD annotation into a SUD annotation is necessarily approximate. The lack of information is particularly problematic in four cases:

1. when several function words depend on the same lexical word in UD (see Section 2.2),
2. when a UD dependency from a lexical word to a function word has to be reversed, some of its dependents has to be transferred to the function word but there is usually no indication on which dependents have to be transferred,
3. to decide whether left dependents of a coordination head are dependent of the whole coordination or of the head alone,
4. when idioms have an internal structure, which is not represented in UD and cannot be recovered in the conversion.

For the first problem, we assume that the further a function word is from the content word, the higher it is in the dependency structure, but there are cases that cannot be solved by such an heuristic, as shown with auxiliaries in Wolof and German (Section 2.2), and our conversion necessarily produces errors without a language-by-language customization.

For the second problem, we have implemented some rules for specific cases: for instance, the subject moves to the auxiliary, while the complements stay on the lexical verb. For modifiers, it is more complex and we resort to word order, preserving the projectivity as much as possible, but only a language-specific and lexicon-based conversion could ensure a perfect structure.

For the third problem, we use heuristics to decide. For example, if the leftmost conjunct of a coordination has a subject to its left and the other conjuncts have no subject, we consider that the subject is shared by all conjuncts.

For the fourth problem, UD flat structures of idioms are converted into SUD flat structures.

4.2 Customization of the UD ⇒ SUD conversion

We have presented default solutions that minimize errors in the UD ⇒ SUD conversion. By customizing the GRS for specific languages, we can further reduce the errors.

For the case of several function words depending on the same lexical word, our architecture allows us to attribute a feature level to dependencies being to reverse with a value that gives its priority in the reversing process. For instance in French, `cop` dependencies are assigned a bigger priority than `aux` dependencies, which means that in case of competition `cop` dependencies must be reversed before `aux` dependencies. Such a rule is needed when the predicate has been extracted as in Fig. 9.

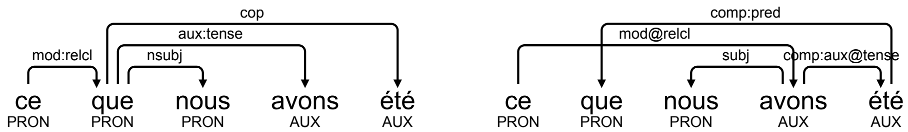


Figure 9: UD (left) and SUD (right) trees for *ce que nous avons été* ‘what we have been’

For the moment, the UD ⇒ SUD conversion has been customized for French and Wolof. For French, a lexicon of modifiers that must move to the auxiliary has been developed. For Wolof, the level mechanism is used to take into account the case described in Section 2.2.

5 The conversion SUD ⇒ UD

Since SUD is richer than UD, we should have no difficulty in designing a universal GRS that converts any SUD annotation of a corpus in any language into an UD annotation. This is globally true but conversion sometimes requires adaptation to the specificity of the language.

The universal SUD ⇒ UD GRS must perform the same tasks as the universal UD ⇒ SUD GRS (see Section 4.1), but in the opposite direction, and the rule order is not the same. It currently contains 94 rules grouped into 20 packages.

In UD, the label of a dependency takes into account not only the syntactic function realized by the dependency but possibly the POS of the governor and the POS of the dependent. For example, the SUD `mod` dependency is converted into a UD `advmod`, `amod`, `nmod`, `obl` or `advcl` dependency, and knowledge of governor's and dependent's POS does not always identify the dependency label. In specific contexts, some words are not used in their usual syntactic function and this use depends on the language.

For example, a SUD `mod` dependency from a verb to a noun is by default a UD `obl` dependency, but there are exceptions. Examples (2) from UD-ENGLISH-GUM illustrate respectively the two cases.

(2) (a) ***Many times*** *prideful people have a serious 'my-way's-the-only-way' attitude.*
 (b) ***An undistinguished student and an unskilled cricketer****, he did represent the school.*

In SUD, the dependencies *have* → *times* (2a) and *represent* → *student* (2b) are both `mod` dependencies. The first one becomes an `obl` dependency in UD, whereas the second one becomes an `advcl` dependency because the noun phrase *an undistinguished student and an unskilled cricketer* is considered as a clause with an ellipsis equivalent to *being an undistinguished student and an unskilled cricketer*.

Since there is no universal criterion to distinguish the two cases, we have designed a SUD ⇒ UD conversion rule, which transforms `mod` relations into `advcl` relations if the governor is a verb and the dependent is a non-temporal nominal preceding the verb, but such a rule only works for certain languages, French and English in particular. Since the rule requires distinguishing temporal nominals, we chose to link the conversion rule to a lexicon. Another solution would have been to mark temporal nominals in the corpus (as it is done in some treebanks with the `tmod` extension).

Another difficulty in the SUD ⇒ UD conversion is that the definition of some UD relations takes into account semantic properties. In particular, the relation between a verb and an argument clause is denoted `xcomp` if the subject of the object clause is controlled by the main verb. Otherwise, the relation is denoted `ccomp`. Consider the following examples extracted from the FRENCH-GSD corpus.

(3) (a) *les mesures visant à développer l'accord* 'measures (aiming) to develop the agreement'
 (b) *Le tourisme commence à se développer.* 'Tourism is starting to develop.'

The UD annotation of (3a) includes a *visant* −[`ccomp`]→ *développer* dependency, whereas the UD annotation of (3b) includes a *commence* −[`xcomp`]→ *développer* dependency. In SUD, both dependencies are denoted `comp:obl` according to the fact that the definition of syntactic relations is based on positional paradigms (see Section 2.1). To choose between `xcomp` and `ccomp` in the SUD ⇒ UD conversion of these relations, a way is to use a lexicon of control verbs and a conversion rule, which uses this lexicon. A major drawback is that you it should be done for each language separately. To avoid this drawback, another way is to mark the relations of the control verbs to the concerned argument with a special feature. That is what is done with the extension `@x` in the SUD annotation.

The method we just described for improving the UD annotation resulting from the conversion can be used to take into account the idiosyncrasies of some languages. The diverse interests behind treebank development regularly lead to some idiosyncratic enrichment of the annotation. UD responds to this need with the option of adding language (or treebank) specific subrelations and features, and SUD naturally follows this approach. If and only if the SUD treebank developers have added new subrelations or features and want them to be taken into account when translating to UD, they must add these idiosyncratic rules to the universal SUD ⇒ UD GRS.

For the time being, the SUD ⇒ UD conversion has been customized for French (by inserting two rule packages in the universal GRS), Naija, and Beja. For Beja, which is a strongly head-final language, coordinations have been analyzed in SUD by head-final `conj` relations (see (Kanayama et al., 2018) for a similar analysis in Japanese and Korean). As `conj` relations must always be head-initial in UD, we have added an ad hoc conversion to a `dep:conj` relation, but it is possible to customize the conversion in another way, for instance, by reversing the direction of `conj` relations.

On the train part of SUD_FRENCH-GSD, the language-specific customization fixes 1.2% of the 400,220 dependencies in the UD ⇒ SUD direction and 0.4% in the other direction (i.e threes times less, which is not surprising). The low percentage shows that idiosyncratic customization can be ignored at first when starting a SUD treebank as the universal SUD ⇒ UD conversion amply does the trick.

The lack of gold annotation in UD and SUD does not allow a direct evaluation of our SUD ⇒ UD and UD ⇒ SUD conversion tools, but we have done an indirect evaluation, using double conversion. The SUD ⇒ UD conversion followed by the UD ⇒ SUD conversion on the SUD_FRENCH-GSD corpus gives 6231 different dependencies out of 400,220 dependencies, i.e. 1.56% of the total, between the resulting annotation and the initial annotation. The UD ⇒ SUD conversion followed by the SUD ⇒ UD conversion on the UD_FRENCH-GSD corpus gives 90 different dependencies out of 400,220 dependencies, i.e. 0.02% of the total, between the resulting annotation and the initial annotation. This highlights that SUD is richer than UD. A closer look at the differences in the first double conversion shows that 82% are due to the flattening of idiomatic structures in UD, the rest coming from the ambiguity of UD in the dependencies on coordinations and nuclei.

6 A practical guide for the development of a SUD treebank

Several tools are already available for helping the start of a new treebank in SUD.

GREW-MATCH (Guillaume, 2021) is an on-line graph query tool which is dedicated to linguistic structures and in particular dependency graphs. It can be used during annotation in order to have a transversal view on already annotated data which helps to take consistent decisions on new annotations. During the maintenance of the corpora, it also helps to ensure global consistency and to do error-mining. GREW-MATCH can be easily coupled with the two UD ⇔ SUD conversion systems and gives access to the parallel view of both annotation schemes: you can search in SUD and see also the UD corresponding structure and the reverse.

The whole annotation process can be managed through the ARBORATORGREW[18] annotation platform (Guibon et al., 2020): user handling, access control, manual edition of the data... GREW-MATCH requests are also available through the ARBORATORGREW platform and detected inconsistencies can be corrected directly. In ARBORATORGREW, the user have also access to some specific tools:

- A lexicon-based view of the treebank for detecting inconsistencies in the annotation of the different occurrences of a form or a lemma

- Automatic graph transformation for the correction of regular errors or for applying changes in the annotation decisions (in the sentence-based as well as in the lexicon-based view of the treebank)

A validation page for SUD treebank is available through GREW-MATCH. It checks that structures are well-formed and helps keeping consistent decisions during the annotation process. Through the conversion to UD, the validation of the UD data adds another layer of verification. Comparing the output of the double conversion SUD ⇒ UD ⇒ SUD with the original data is an additional way to obtain valuable feedback on the annotated data.

It should be noted that in the particular case where a UD treebank already exists, the universal conversion should be tested to verify that the internal structure of the nuclei matches the expected structure. If this is not the case, the conversion may need to be customized as explained in Section 4.

7 Conclusion

SUD is not just a richer and easier annotation scheme than UD that can automatically be converted to UD. Importantly, SUD's distributional criteria facilitate and homogenize the annotation choices, resulting in treebanks that enable typological measures across languages. Also, a rich set of tools is available that allow for a kick-start in annotation of raw or partially annotated data. Several SUD treebanks exist that can serve as examples, with more in the pipeline. Go SUD!

[18] `https://arborator.github.io`

References

Olivier Bondéelle and Sylvain Kahane. 2021. Les particules verbales du wolof et leur combinatoire syntaxique et topologique. *Bulletin de la Société de Linguistique de Paris*, 115(1):391–465, January.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, 07.

Matthew S. Dryer. 1992. The greenbergian word order correlations. *Language*, 68(1):81–138.

Kim Gerdes and Sylvain Kahane. 2013. Defining dependencies (and constituents). *Frontiers in Artificial Intelligence and Applications*, 258:1–25.

Kim Gerdes and Sylvain Kahane. 2016. Dependency annotation choices: Assessing theoretical and practical issues of universal dependencies. In *LAW X (2016) The 10th Linguistic Annotation Workshop: 131.*

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2018. SUD or surface-syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 66–74, Brussels, Belgium, November. Association for Computational Linguistics.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2019. Improving surface-syntactic Universal Dependencies (SUD): MWEs and deep syntactic features. In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 126–132, Paris, France, August. Association for Computational Linguistics.

Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. 2020. When Collaborative Treebank Curation Meets Graph Grammars. In *LREC 2020 - 12th Language Resources and Evaluation Conference*, Marseille, France, May.

Bruno Guillaume. 2021. Graph Matching and Graph Rewriting: GREW tools for corpus exploration, maintenance and conversion. In *EACL 2021 - 16th conference of the European Chapter of the Association for Computational Linguistics*, Kiev/Online, Ukraine, April.

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

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

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey, May. European Language Resources Association (ELRA).

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

Peter Timothy Saunders. 1980. *An introduction to catastrophe theory*. Cambridge University Press.

Karel Van den Eynde and Piet Mertens. 2003. La valence: l'approche pronominale et son application au lexique verbal. *Journal of French language studies*, 13(1):63–104.

Daniel Zeman. 2008. Reusable tagset conversion using tagset drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA).

Mutual dependency and Word Grammar: headedness in the noun phrase

Nikolas Gisborne
Linguistics and English Language
School of Philosophy, Psychology and Language Sciences
University of Edinburgh
n.gisborne@ed.ac.uk

Abstract

Most Dependency Grammars reject mutual dependency but Word Grammar allows it, with Hudson (2004) exploiting it to analyse Determiner+Noun constructions. I discuss whether mutual dependency is desirable in Word Grammar and whether it is necessary in the treatment of Determiner+Noun structures. Many Dependency Grammar theories assume that the common noun is the head; early Word Grammar (Hudson, 1984) on the other hand argued for Determiner as head, and only more recently for mutual dependency. Mutual dependency is not permitted in most Dependency Grammars for formal reasons, because it violates the usual acyclicity constraint. However, natural language requires some relaxation of formal constraints on representations. I argue against mutual dependency in Word Grammar and also argue that it is not necessary in the analysis of Determiner+Noun; however, my arguments come from within Word Grammar and are based on its cognitive assumptions about how grammar is represented in the mind, rather than being based on formal criteria. My aim is to show that within a cognitive theory, constraints on representations can and should be stated in terms of the nature of the human cognitive system.

1 Introduction

Within a grammar that rejects exocentric analyses, there are three possible syntactic structures that can be assigned to the phrase *the dog*: (i) *the* depends on *dog*; (ii) *dog* depends on *the*; and (iii) the words are mutually dependent. The first two choices make no particular or unusual claims about the nature of Dependency Grammars (DGs): DGs are a class of grammar where there are pairwise relations between words. Both (i) and (ii) are compatible with constrained DGs and neither introduces formal violations of DG architecture. On the other hand, (iii) violates most dependency architectures because, apart from Word Grammar (WG), DGs adopt an acyclicity constraint.

Robinson (1970, 260) presents a series of axioms for a DG, given in (1). She states that the dependency relation is 'transitive, irreflexive, and anti-symmetric'. She gives these as the 'axioms of the theory which was advocated by Tesnière (1953), (1959) and formalized by Hays (1964) and Gaifman (1965).'

(1) a. one and only one element is independent;
b. all others depend directly on some element;
c. no element depends directly on more than one other; and
d. if *A* depends direct on *B* and some element *C* intervenes between them (in the linear order of the string), then *C* depends directly on *A* or *B* or some other intervening element.

Robinson's axioms (1a-c) define a DG as a rooted tree, with (1d) also enforcing projectivity. Not every dependency theory subscribes to all of these axioms, but they offer a starting point. Ballesteros and Nivre (2013, 6) and McDonald and Nivre (2011, 202) describe similar constraints, with the latter telling us that

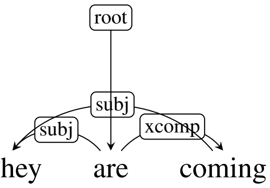


Figure 1: Analysis of *They are coming.*

the requirements in (1a-c) are ‘consistent with most formal theories’, such as Functional Generative Description (Sgall et al., 1986) and Meaning-Text Theory (Mel’čuk, 1988). The theories underwriting Osborne (2019) and Järvinen and Tapanainen (1998) also fit. McDonald and Nivre (2011) identify Word Grammar as a theory which is an exception in that it relaxes (1c), as does Anderson (2006)’s theory.

Figure 1 shows multiheadedness in a Word Grammar analysis of *they are coming*, using the analysis of Hudson (1984) and (1990). It represents *are* as the root of the sentence and *they* as the subject of both *are* and *coming*. Languages with agreement between syntactic subjects and predicative participles show why there has to be a syntagmatic relationship between *coming* and *they* in *they are coming*. In the absence of a syntactic relationship, what would, or could, carry the agreement in an example such as the French *elles*-FEM.PL *sont venues*-FEM.PL? As a result of this multiheadedness, if WG did not have mutual dependency, its representations would be directed acyclic graphs rather than dependency trees. However, as I have said, WG allows mutual dependency as well, including between Determiner (D) and Noun (N), making its representations general graphs. It is the point of this paper to show that this causes problems for WG theory, even given WG’s cognitive architecture, and that mutual dependency is not necessary in the case of D+N (Hudson, 2004).

Returning to the axioms in (1), it is worth briefly noting that many theories reject the projectivity constraint in (1d). For example, Mel’čuk (2014, 21) points out that the Latin sentence in (2) is non-projective due to the discontinuous relationship between *meas* and *nugas*.

(2)	Tu	*solebas*	**meas**	esse	aliquid	*putare*	**nugas**
	you-NOM	used-2SG	my-FEM.PL.ACC	be-INF	something-NOM	think-INF	trifles-FEM.PL.ACC

‘You used to think that my trifles were something’

The dependency between *solebas* and *putare* crosses the dependency between *meas* and *nugas* in a violation of projectivity. Many natural languages have such projectivity violations built in and one of the advantages of a non-projective DG is that it allows a direct, surface-level, representation of the syntactic structure of languages with discontinuous word order.

Constraints on WG representations are discussed in Hudson’s monographs. In Hudson (1984, 98-9) there is the Adjacency Principle, combined with the Priority to the Bottom Principle, which allows non-projectivity in extraposition; in Hudson (1990, 144ff.) there is a simplified version of the Adjacency Principle, but revised to allow multiple heads, retaining a version of Priority to the Bottom; in Hudson (2007) there is a theory of word order that dissociates dependencies from landmarks, with landmarks being responsible for word order. In Hudson (2010) this theory is developed and refined so that landmarks have to be projective, not dependencies themselves. Therefore, in WG projectivity is revised to allow extraction and, as we have seen, the constraint in (1c) is relaxed to permit structure sharing in predicative complementation, raising and control structures.[1]

What about mutual dependency, which Hudson (1990, 197) introduces in his analysis of relative clauses, and which is disallowed in other dependency theories? To the best of my knowledge, there are no mutual dependency analyses in Hudson (1984), or the papers that appeared between Hudson (1984) and Hudson (1990). I also have not found any particular arguments for mutual dependency: as far as I can see, it was adopted without much discussion, although the argument would be that looping

[1] Although I should note that Creider and Hudson (2006) introduce covert words into the WG ontology to handle a subtype of the infinitival construction in Ancient Greek, which introduces a further difference between WG and other DGs: as a consequence of this move, WG allows pairwise dependencies between realised and unrealised words.

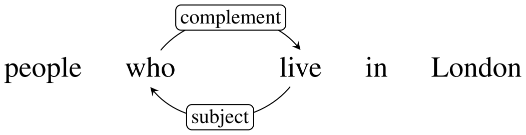


Figure 2: Mutual dependency in syntax

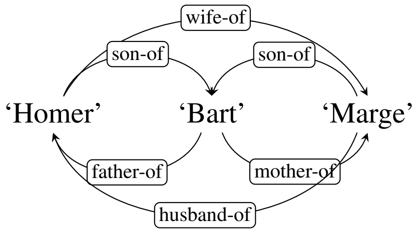


Figure 3: Marge and Homer: husband and wife;
Bart has two parents

structures are inherent in networks, and in cognition. Hudson's representation for the phrase *people who live in London* (Hudson, 1990, 197) is given in Figure 2. Should structures like the one in Figure 2 be allowed in the syntax? This depends on the premises underlining the theory, and whether such structures are consistent with how the theory works.

WG is explicitly a cognitive theory (Hudson, 1984, 31-35) where language is analysed as part of a mental/cognitive network. Mutual dependency is a type of loop, and loops clearly exist in both network structures and mental networks. Hudson (2010, 49) makes this clear in his representations of family structure such as the partial structure of the Simpsons' family in Figure 3, where he shows the mutual relationship between Homer and Marge, and between each of them and Bart.[2] The point is that such structures are simply part of how a cognitive network must be structured.

The diagrams in Figures 2 and 3 are partially simplified, because in a full WG network they would also show classification relationships. The WG network is a classified network, with both nodes and arcs classified, where 'isa', the predicate of default inheritance, is a primitive. Given the network approach to cognition, the question for WG is whether mutual dependencies are cognitively plausible, not whether they meet a set of graph-theoretic constraints. The cognitive network approach to language permits the researcher to relax formal constraints, as long as the architecture is compatible with known properties of the mind and brain. This does not absolve the researcher from embedding the research programme in a set of constraints that limit possible theories. It means that the game is different: the constraints are not those of formal language theory; instead, the theory is obliged to be bound by findings from cognitive psychology and related areas. WG has a dependency theory of syntax because it is the syntactic theory that is compatible with the network theory of language and cognition. A dependency graph is a network with terminal nodes, so it is consistent with the idea that language is a cognitive network. For a cognitive theory, the issue comes down to two things: (i) is the structure parsable by the *human* parser? And (ii) is it learnable? To learn a grammar, the speaker/hearer will start small and build the grammar incrementally, learning words and dependencies, which can be learned from adjacent words. There is flexibility built in. Once learned a dependency can be subject to processing variations, even interruptions: *I was wondering—could you pass the salt please—whether you'd ever been to Italy*.

Therefore, for Hudson, mutual dependency is acceptable in syntax, because loops are found in ordinary relationships in cognition and loops are therefore learnable. In the rest of this paper, I argue that their learnability in the general case does not mean that loops should be tolerated in syntax. In particular, I argue that syntax is different from the rest of the cognitive network, for three reasons. The first is that while networks are non-directional, syntax is necessarily directional. The second is that syntax

[2] The lack of a 'root' and the single quotation marks around *Homer*, *Bart* and *Marge* show that this is a fragment of conceptual structure, not syntax.

necessarily involves terminal nodes, and therefore generating or parsing sentences always involves active searching for and retrieving information in an inheritance hierarchy. The third difference between syntactic dependencies and other cognitive relations is that dependency is a hierarchical relationship between head and dependent. These properties make dependencies conceptually richer than other network relations, and introduce the possibility of conflicting information in mutual dependency. In Section 2, I address these issues, then in Section 3 I re-examine Hudson's arguments for mutual dependency within the NP.

2 Why is syntax different?

The WG cognitive network is claimed to have the small-world and scale-free properties (Steyvers and Tenenbaum, 2005) found in semantic networks which are also found in dependency networks (Ferrer i Cancho and Solé, 2001; Ferrer i Cancho et al., 2004). This is unsurprising: dependencies are a particular subtype of grammatical relation, and grammatical relations are a subtype of the relations we find in cognition. However, as a subtype they are conceptually richer than the network relations that they inherit from. Dependencies are implicated in both form and meaning. They establish hierarchical relations among words. They are responsible for word order, combine forms, trigger the morphophonological facts of agreement, and they combine referents of words with the semantics of the heads of those words. We therefore need to be cautious about presuming that the gross similarities between dependencies and other cognitive relations mean that syntax is just like the wider cognitive network. Furthermore, there are different subtypes of dependency. Some, but not all, are associated with the landmark relation and are responsible for word order. And some, but not all, are valents.

Linearisation is self-evidently a key element of syntax so we can take it first. We can say *the dog* but not **dog the*; *in the kitchen* is part of English but **the in kitchen* is not. In this respect, syntax is different from the network structures elsewhere in cognition. But it gets more complicated. Take *La table que j'ai achetée est là*, 'The table I bought is there', where *achetée* agrees with *La table*. In French, perfect constructions with AVOIR do not usually trigger agreement between the participial complement and its subject, unlike those constructions with ÊTRE. There is no agreement in *Elles ont acheté la table*. However, when the participle's direct object is linearised before it, agreement is triggered between the direct object and the participle. This is not just a prescriptive rule: the agreement can also be heard as in *la lettre que j'ai écrite*. See also Cong and Liu (2014) and Liu et al. (2017) on linearisation.

Because syntax has to involve terminal nodes, it also involves both the retrieval of taxonomic information and a process of updating it. The rest of the cognitive network will also involve terminal nodes, because it will be constantly updated in the face of new information, from different types of perceptual information including speech perception, but syntax always involves terminal nodes because it involves the classification of utterances, which must be linearized, and which are related to the conceptual-intensional system of semantics. In WG, each word is an action with a time, place and speaker (=actor); each of these actions is linked to a concept in the conceptual-intensional/semantic part of the language network. The database of information that makes up the permanently stored mental network is a database of '*declarative* knowledge, expressed as propositions' (Hudson, 1984, 2; emphasis original). This database of propositions licenses the utterances or allows the hearer to form inferences about the structure and determine a parse for what they hear, by exploiting default inheritance as they classify each utterance token. Word Grammar is a constraint-based theory and WG syntax is an interface between the stored database of propositional knowledge in the conceptual network, and the human behaviour of making and interpreting utterances. It involves a mental network sitting on top of a neural network, with undirected spreading activation in the network (Collins and Loftus, 1975). For mutual dependency to be possible, it has to be consistent with the constant searching for, and updating of, information that makes up real-time language use.

The hierarchical nature of syntax is baked into dependency analyses: dependency is a pairwise relationship between words where one is the head and the other is the dependent. Heads have certain properties: they are responsible for distribution, and also for the internal structure of a phrase.[3] For

[3] Within WG, Rosta (2006) has argued for factoring these notions out and contended that they are distinct in order to allow

example, in *De officiis* (Latin, 'On duties') *de* selects a noun, determines its position and governs its case (ablative). The preposition is clearly the head, *officiis* the dependent. In the case of mutual dependency, it is hard or impossible to know which word has chosen the other, which word is responsible for the form of the other, and which word is responsible for the (relative) position of the other. The selection and positional information is both hierarchical and linear: a word of type *x* requires a word of type *y* to its right or left and, in languages with the right kind of morphology, governs its form. The point about syntax being comprised of terminal nodes is that in WG this information has to be looked up in the inheritance hierarchy. It therefore has to be stored as a generalisation that words of type *x* have these properties. Furthermore, the hierarchical nature of syntax is involved at a depth of complexity: in *he*$_i$ *made the food for himself*$_i$, the word *himself* is linked to *he* as a co-dependent (in some relevant sense) of the same head.

What do these facts mean for mutual dependency, given the analysis in Figure 3 and the observation that loops must exist in conceptual structure? It means we can accept that loops must exist in the symbolic network, while simultaneously requiring there to be further supporting theoretical evidence for mutual dependency in the syntax. We know that the nature of cognition places constraints on syntactic structure: there is a literature exploring how working memory constrains dependencies, particularly dependency distance, (Cong and Liu, 2014; Futrell, 2017; Futrell et al., 2020; Gibson, 1998; Gildea and Temperley, 2010; Liu, 2018; Liu et al., 2017) and it is appropriate to think about the relationship between working memory, language dynamics and language structure. In the case of dependency distance, Cong and Liu (2014, 605) cite spreading activation as key: 'Given the small-world topology, whereby each pair of vertices can be generally connected by a short path, the loss of activation energy can be minimized and the success of retrieval is thus maximized.' As they say, this gives rise to a preference for minimal dependency distance, which is a constraint on structure consequential upon the interaction of linear information flow and working memory limitations. What consequences do such facts have for the status of mutual dependency?

For there to be mutual dependency, it must be consistent with the design features of the cognitive network. It is with some of them: a mutual dependency relationship has a small-world structure which, following Cong and Liu (2014), should facilitate retrieval and minimise the loss of activation energy making the structure easily chunked and passed into longer-term storage. But this is not the whole story because there are other considerations. The specifically unique properties of syntax, its hierarchical, linear structure, make it possible for mutual dependency to be coherent to the extent that syntax fits a network topology, but not coherent in respect of hierarchy and linearity, or inheritance. The way to address this is by asking whether mutual dependency might cause problems for a WG analysis.

It does. A hierarchical, linear structure places constraints that do not apply in an ordinary network. In the case of the D+N construction, Hudson (2004, 32) writes, 'Since we have seen that D and N depend on each other, either of them can be the head of the NP, and the choice can be left to the surrounding construction.'[4] Therefore, either word is the head, depending on the surrounding linguistic context. Within the phrase, with mutual dependency, both words are landmarks for the other, and select the other. These two consequences render the phrase effectively headless: D+N does not have a formal structure where we know that D, or N, is the head, responsible for the distribution of the phrase, selecting the other element, determining the linear position of the other, and governing the other element's form.[5]

There are two possible consequences for such a theory. The first is that the headedness of D+N is unknown until the construction containing the D+N string resolves it, in which case D+N is unique among the major, frequently attested, constructions of English in being both headless and constructional. This would mean that the speaker/hearer would not know which word was the head until the constructional context determined it, creating a problem for eager parsing in the word-by-word approach to dependency parsing of Covington (2001). Here, as each word is encountered, it is attached to a classification which

for a range of phenomena such as pied-piping. For Rosta, in the grammar of English, the structure of a phrase is mainly decided by evidence such as word order and ellipsis. Distribution concerns the positions where a phrase may occur.

[4] The thin entering wedge for this position is Hudson's treatment of determiners as (transitive) pronouns, a position he adopts rather than Postal's analysis that pronouns are determiners. Given that pronouns are in turn nouns, both elements in D+N are nouns, and therefore hypothetically either may serve as head.

[5] Mutual selection is not inherently a problem; it is determining the head for distribution and structure that is. Mutual selection is indirectly a challenge, however, because it contributes to the problem of identifying the head.

tells the speaker/hearer what to expect about its behaviour. Perhaps at best the internal structure of a mutual dependency phrase could be resolved. But on this approach, neither word has an incoming dependency relating it to its head until that is determined by the containing construction—which could be to the right of the D+N string at issue. This means that the D+N string has to be held in memory until its head can be determined. But processing is rapid, and anticipatory, and has to happen in the context of Christiansen and Chater (2016)'s 'Now-or-Never bottleneck', which is a working-memory constraint on linguistic production. Due to the fleeting nature of memory, the brain has to compress and recode linguistic information fast and efficiently, across all of the levels of grammar, otherwise the information is lost. Anticipatory parsing is a consequence of the bottleneck; it requires linguistic units to be chunked and then passed up to longer-term storage as the information flow moves along. It is not possible if we do not know which word is the head until the wider context tells us.

The other possible consequence is that there exist two propositions, 'D is head' and 'N is head', and the speaker/hearer has to resolve them. The problem with the two proposition approach is that it involves a contradiction. Like everything else in the cognitive ecology, headedness is learnt on the basis of experience. To learn mutual headedness in syntax would need a model where it was not contradictory. A contradiction can only be resolved by stipulation, otherwise it gives rise to a failure of structure such as with the Nixon diamond in multiple inheritance (Touretzky, 1986) which, Hudson (2000) argues, gives rise to ungrammaticality, and accounts for the lexical gap of *amn't*. Grammars are (by and large) regular, coherent and learnable, which requires simple structures that facilitate rapid structure building in the online work of parsing and production. If contradictions render lexical gaps, this contradiction should render a mutual dependency analysis of D+N impossible, and in turn, if a mutual dependency analysis were required, there should be a constructional gap: *D+N.

There are two possible routes out of this pair of problems. One is to adopt the approach of Rosta (2006) mentioned in Footnote 3 and to factor out different dimensions of headedness. As long as only one word is the structural head, and only one the distributional head, this approach will work for D+N. Another is to allow mutual syntactic relationships, but to make only one of those relations a dependency, with the other relation carrying a depleted degree of syntactic information. Such an approach is permitted by the network topology and consistent with Hudson's theory of landmarks for word order. Moreover, by making only one of the words concerned responsible for distribution and non-conflicting aspects of the internal structure of a phrase, this approach avoids the disadvantages of mutual dependency. However, the evidence is clearly that mutual dependency is a problem for Word Grammar, for all that other kinds of loop in the syntax need not be. I now turn to the Hudson/van Langendonck arguments about headedness of the D+N construction. I argue that the arguments Hudson offers in favour of N do not have sufficient force that we are obliged to adopt a mutual dependency analysis of this construction.

3 The head of D+N

In this section, I concentrate on the arguments that Hudson (2004) adduces in favour of mutual dependency of D+N. I do not address his earlier arguments that D is the head, nor do I address other arguments in the literature about the headedness of D+N such as those in Osborne (2021). My concern in this section is in establishing whether the five arguments Hudson (2004) presents in favour of N as head are sufficient reasons to adopt a mutual-dependency analysis of D+N. The shape of the argument is that if his evidence that N is (also) the head can be adequately challenged or is inconclusive, there is no reason from the data to adopt mutual dependency.

We can begin by thinking about what determiners do. In (3), there are a number of nouns of different kinds, serving as the subject of the main verb, with various constraints shown.

(3) a. Ovid was banished to Tomis.
b. She is in the sitting room
c. Water flowed out of the kitchen door.
d. Students poured out of the classroom.
e. *Dog trotted down the street.
f. *The my dog trotted down the street.

g. John's dog trotted down the street.

In (3) we see that proper nouns, pronouns, mass nouns and plural count nouns can all occur without a determiner (3a-d), but singular count nouns cannot (3e). The example in (3f) tells us that it is not possible to have chains of determiners in English. And (3g) tells us that possessives also make a singular count noun grammatical. These restrictions are language specific: for example, proper names in Ancient Greek have articles and Latin allowed examples such as (3e). The theory, then, must be a parochial account of English. Finally, the subjects in (3a-d) are all referential. Proper nouns and pronouns are necessarily referential, and the evidence of (3c-d) is that mass nouns and bare plurals are weakly existential, but bare singular count nouns are not. One of the things that determiners do is to fit singular count nouns up to refer. Another is that they make the reference of other nouns more precise (or determine their reference).

Hudson (2004) argues that mutual dependency is necessary in the analysis of Determiner+Noun strings because of various arguments in favour of N as head published in Van Langendonck (1994). Given that Hudson (2004) continues to find his own earlier arguments that the Determiner is the head compelling, he concludes that there is a state of mutual dependency between the determiner and the noun, and that the headedness of the construction is moot. The most straightforward way to tackle whether mutual dependency is necessary in a WG analysis of D+N is to assume that Hudson's earlier analysis will work, and to explore how compelling the additional arguments for N as head turn out to be on a second look. If it is possible to dispose of the new arguments in Hudson (2004) then it is possible for the theory to dispense with mutual dependency in this area of grammar by reverting to the earlier analysis. Hudson (2004) relies on Van Langendonck (1994) for arguments that N is head as well as Osborne (2003) and Huddleston and Pullum (2002).

The main arguments in Hudson (2004) that D depends on N are: (i) NPs as adjuncts; (ii) Possessives (3g); (iii) the need for determiners (3d); (iv) the single determiner constraint (3e); (v) facts about extraposition. I take these in turn.

NPs as adjuncts

The relevant examples are shown in (4), taken with the quotations from Hudson (2004, 11); he takes this as the most important set of facts in Van Langendonck (1994).

(4) a. I saw him *this morning*
b. It's best to do it *my way*
c. Put it *this side of the line*

Hudson's claim is that the 'NPs that can be used in this way are defined exclusively in terms of their N; the D is more or less irrelevant, being freely selected according to the normal rules.' There are two further key restrictions: the italicized NPs in (4) cannot be replaced by a personal pronoun or *this* without a complement, and '[A]lthough all the eligible nouns all refer to times, places and manners, they are also lexically quite restricted': for example, it is possible to use WAY in these constructions but not its (near) synonym MANNER. See (5).

(5) a. *I saw him it.
b. *It's best to do it mine
c. *Put it this.
d. I did it the usual way/*manner.

A further restriction is clear with NP time adjuncts: we can say *I saw him this morning* but not **I saw him this party* even though PARTY can refer to a time in an expression such as *I saw him before the party*. However, it is unclear whether this is a separate lexical restriction, or whether it there is a semantic generalization. Perhaps PARTY is excluded because its basic semantics is to refer to an event, and it refers to the time of the event by metaphorical extension. So how should we account for these facts?

The examples are time, place and manner adjuncts and the key facts about adjuncts are that they are not selected; they define their own semantic relation to their head; and they 'reverse unify' with their heads. They are syntactic dependents, but in the semantics they take their heads as their arguments.

Hudson claims that these D+N constructions should be treated as having the N as the head because it appears to decide their ability to occur as adjuncts. However, there is a further set of examples that he introduces that make the situation more complicated. Adjunct D+N patterns also place restrictions on the determiners that occur in them, (6).

(6) a. He did it this morning/*the morning/*a morning.
b. He did it this way/*the way/*a way.

As (6) shows, THE and A cannot occur in these adjunct NP constructions, unless the N is further modified: *he did it the right way* and *he did it the same morning* are both fine. The restrictions are also related to the noun in the construction: as Hudson (2004, 12) points out there is variation among the nouns that can occur in adjunct NP constructions in terms of which determiners they occur with.

(7) a. I'll do it in my (own) time
b. I'll do it on my day.
c. *I'll do it my time/day.

Although TIME and DAY can occur in NP adjuncts (*I saw him this time/that day*), and can occur with possessives within PP adjuncts, they cannot occur with a possessive when they function as NP adjuncts.

The final set of facts discussed by Hudson has to do with relative clauses. He notes that the restrictions on nouns in adjunct NPs survive relativisation, where the noun is the external head of the relative clause, as in (8) (Hudson, 2004, 13). In the examples, '*way* is possible but *manner* is not, and *time* is possible but *point(in time)* is not.'

(8) a. The way/*manner he did it shocked us.
b. I remember the time/*point he did it.

Hudson (2004, 13) uses the relative clause structure in (8) as an argument that the D is irrelevant. However, there is a complication that Hudson does not discuss: although (9a) is fine, none of the examples in (9) are acceptable.

(9) *My/*this/*that way he did it shocked us.

To summarise the restrictions: only nouns with the right semantics can appear in these adjunct constructions; and only certain nouns can appear in the construction with their apparent synonyms being excluded; there are environmentally conditioned restrictions on the determiners that occur in these constructions.

It is worth noting that the restriction in (6) co-varies with whether the noun is modified or not. The relative clause restriction in (8) provides some evidence, but also they restrict modification in examples such as **I saw him this inconvenient time*. However, *I saw him that terrible day* is fine.

It is not obvious that these facts argue against the traditional WG analysis of D+N, with D as head. The restrictions on nouns such as MANNER and different determiners resemble those with *kick the bucket*.

(10) a. He kicked the bucket/*a bucket/*this bucket/*some bucket.
b. *He kicked the pail/scuttle/pitcher.

Just as in idioms, there are restrictions on both N and D in the NP adjuncts: it appears that NP adjuncts are idioms. How should we treat idioms? There is a WG analysis in Gisborne (2020, 44-55) which we could adopt. The analysis relies on WG's default inheritance architecture: in the case of *kick the bucket*, there is a special subtype of the lexeme KICK, which selects a special subtype of THE which in turn selects a special subtype of BUCKET. That is to say that the whole string is defined as a regular collocation, with a particular meaning, but the normal syntax of the phrase is maintained with *the* still the head. A similar analysis of NP time and manner adjuncts would help capture the degrees of irregularity we see in these examples. We know, both from the discussion in Gisborne (2020, 44-55) and also from Nunberg et al. (1994) that there is potentially a great deal of variation within the expression of different idioms.

Given the apparently arbitrary restrictions on NP adjuncts, they invite a similar analysis where the various lexical restrictions—on both D and N— are treated as reflexes of the construction's idiomaticity. The appropriate analysis would then be to treat the different variants as subtypes (sublexemes in WG's terms) of an NP time or manner adjunct type in the inheritance hierarchy.

The place adjuncts in particular argue in favour of this analysis. *Put it this side of the line* requires the mention of a (semantic) landmark relative to the line: *Put it my/your/his/this/that side of the line* are all fine, but **Put it the side of the line* is not because the definition of where, relative to the line, is relevant to the definition of place that the adjunct is contributing. This also shows that there are degrees of idiomaticity in this area of grammar, which is consistent with what Gisborne notes for idioms more generally, and is also by and large compatible with Nunberg et al. (1994)

The one argument we still need to discuss is the argument from the interaction of adjunct NPs and bare relative clauses in examples such as (8). Hudson argues that as N is the antecedent of a relative clause, not NP, the structure of (8)(a) is *The [way he did it] shocked us* rather than *[[The way] he did it] shocked us*. The analysis that the relative clause depends on N (or its analogue in phrasal theories) is standard since Partee (1975); the claim that N is the antecedent of the gap is also widely adopted, see e.g. Huddleston and Pullum (2002, 1037). The reasoning is that the determiner determines the whole of the syntagm [N+relative clause], and so D+N cannot be the antecedent of the gap. However, although the attachment of the relative clause is clear, it is not clear that it tells us about the filler-gap relationship, because there is a conflict of facts: bare singular count nouns cannot occur as the dependents where gaps are found. This is easier to see with argument gaps than adjunct gaps, because their positions are fixed. For example, *The party he got drunk at __ was yesterday* requires a determined count noun to fill the gap because **He got drunk at party* is ungrammatical. This implies that he antecedent of the gap is *the party* even though the relative clause *he got drunk at __* modifies *party*, not the whole of the NP: the preposition AT cannot occur with a bare count noun. On the other hand, Sauerland (1998, 65ff.) presents reconstruction evidence for a relationship between the head noun and the gap. The best conclusion is that bare relatives are a topic for further research, not a knock-down argument for N as head.

Possessives

Hudson presents two different arguments from Van Langendonck concerning possessives. The first is evidence from Dutch, which is only indirectly relevant, because Dutch is not English and the grammar of English NPs involves parochial facts about English. See (11).

(11) a. Moeders jurk 'mother's dress'
b. Peters moeders buren 'Peter's mother's neighbours'.

The argument is that (11a) is a hyponym of *jurk* 'dress', so this is the head; however the same is true for a phrase such as *the dog* which is a hyponym of *dog*, where previously Hudson has argued that *the* is the head. Here, I think, we can take a different approach. The key fact is that in Dutch (and German for that matter) a singular count noun cannot occur on its own as in English, but it is not only rendered grammatical by a determiner: a genitive noun, which cannot co-occur with a determiner, can also make it grammatical. Van Langendonck and Hudson's argument is that the genitive noun is dependent, and therefore the determiner must be dependent.

Perhaps this is so. But first, English does not have case, except vestigially on personal pronouns, so the arguments are not directly relevant and the comparison has to be appropriately set up. In fact, there is a comparison which obviates the need to take N as head, which we can find by thinking about what determiners do. In an article developing a WG theory of the diachrony of the English definite article, Gisborne (2012) argues that the article is a quantifier expressing both existential and universal quantification over the noun, following arguments in Russell (1905) and Neale (1990). If the purpose of articles is to quantify over nouns, then by extension this is the general purpose of determiners, because the treatment of definite articles extends, as Gisborne discusses, to other definite elements, and because there are already determiners whose role is to quantify. The indefinite article simply provides existential quantification. Philippi (1997) provides interesting evidence for this position in a discussion of the emergence of the article system of German, and also provides further evidence that case can have existential force, thereby quantifying existentially over the noun—a suggestion which is supported by differential case marking. If this is right, we can account for the Dutch data in (11) with a simple observation: the genitive noun is relational. The genitive case gives existential quantification over both the noun that it is attached to and the noun that the genitive relates to. There is no scope for double determination (or quantification) and so the similarity between the Dutch genitive in (11) and English possessive *'s* is captured.

There are further arguments about the synonymy of *the old man's hat* and *the hat of the old man*. Hudson (2004, 16) addresses those arguments. Van Langendonck's main argument is from the idiom *to pull someone's leg*, meaning to tease someone. The observation is that in the idiom, the possessor is required in both variants, and **She pulled the leg* is ungrammatical in the idiomatic sense (although it is perfectly grammatical if it is taken literally, and said about the chicken on the table). This argument has little force, because it concerns the structure of an idiom. In much the same way as we can omit neither the head nor the direct object of *She kicked the bucket* nor change either of them, we cannot rework the structure of this idiom without it losing its idiomatic meaning.

The need for determiners

Hudson presents a number of arguments from the need for determiners. The argument is that if a word is required by another word then it is the dependent of that word, as in the case of valency. Singular count nouns require a determiner to be able to occur in argument positions; without one, if they occur in an argument position they are forced to a mass interpretation: *Dog was all over the road*. Although this looks as if D is required by N, we can take the argument in the previous section and rethink the facts. Let us assume that determiners quantify over nouns. How are nouns quantified over without a determiner? The examples in (3) show that there are different ways in which a noun might be quantified over. Proper nouns and pronouns are inherently referential. Possessives are referential because *'s* is inherently definite. Plurals are weakly existential, because plural marking asserts the existence of more than one of the entities denoted by the noun. Mass nouns are the most difficult to describe in these terms. In examples such as *Water is necessary for life* they are weakly generic. The example in (3)(c) is weakly existential: *Water flowed out of the kitchen door* asserts the existence of the water and it is upward entailing. Generics are not upward entailing, on the other hand. However, it is also different from *Some water flowed out of the kitchen door*, which implies a finite mass of water. I think that this property of mass nouns follows from the nature of massness. Mass nouns refer cumulatively: if I show you a heap of rice, I can say, 'This is rice'; I can then show you another heap and say the same, which I can also say of both heaps. This shows that mass nouns by default presume the existence of the stuff that they denote, and they do not require external quantification, unless we quantify over them partitively.

From this we can conclude that singular count nouns are the only subtype of noun requiring some linguistic formative to assert their existence. In case-marked languages, case itself can do this, as is indicated by differential case marking. For example, in Finnish partitive case gives rise to an indefinite interpretation whereas accusative objects are interpreted definitely. In a language such as English, on the other hand, in the absence of case a singular count noun cannot occur unless it is quantified by a determiner. As the determiner quantifies over the singular count noun, it must be the head.

The single determiner constraint

Only one determiner is permitted in D+N structures. Because heads can only select one of a given type of dependent, Hudson sees this as an argument that D is a complement of N: the argument is that there is a single 'slot' for a determiner in the grammar of a noun, in much the same way as it is only possible for a verb to have a single subject or a single direct object, and so once it is filled no other element can occur in that position. This suggestion has the advantage of capturing the generalisation about Dutch genitives discussed above. But if we exclude examples of nominal modification, such as *the boy actor*, only one noun is permitted which is why **the dog cat* is ungrammatical, and to have two nouns determined by a single determiner, they have to be coordinated: *the dog and cat*.

But the constraint does not only apply to dependents. It also applies to heads: it is also only possible to have one head in a given structure. We can analogise from auxiliary verbs. In a string of English auxiliary verbs, it is only possible to have one finite verb: *She may have gone* is grammatical; **She may has goes* is not. This is simply a matter of selection. Each auxiliary selects the form of its complement.

Likewise, determiners. It is part of the grammar of THE that its complement must be a common noun and not another determiner. Other quantifying expressions such as ALL will permit a definite determiner, on the other hand: *all my teachers; all those dogs; all the cakes*. This constraint can be argued either way, and it is not a knock-down argument for N as head.

Extraposition
The final constraint which Hudson (2004, 20) takes as evidence for N as head is extraposition. The constraint is shown in the example in (12). The argument is that it is only possible to extrapose dependents of nouns which themselves depend directly on the main verb.

(12) a. People [who have been waiting ten years] are still on the list.
b. People are still on the list [who have been waiting ten years]
c. *Names of people are still on the list [who have been waiting ten years].

However, the generalisation in (12) is not the full description. In (13) the extraposition is possible, even though the noun does not depend directly on the verb.

(13) a. All of the people [that have been waiting ten years] are still on the list.
b. All of the people are still on the list [that have been waiting ten years].

The conclusion is that it is possible to extrapose dependents of nouns which do not depend directly on the main verb, and that the data in (12) need a separate explanation. The evidence in (13) does not only undermine the argument that (12) is claimed to demonstrate: it also shows that it is possible to extract a dependent of N when N is itself quantified over. If the earlier claims that D quantifies over N are correct, then (13) suggests that extraposition is fine out of a quantified N, and also that D is a quantifier.

4 Conclusions

Primarily because of the problem it causes in identifying the head, and the problems that this brings about for processing, I have argued that WG should reject mutual dependency, despite its network architecture and cognitive basis. This is not an argument that there should be no syntactic loops; it is an argument that if loops are to feature in the syntax, they should not be found among the dependencies which are relevant to distributional structure. There should be no mutual dependency with each word (potentially) the head of the other and the landmark of the other. I have shown that the arguments of Van Langendonck and others which Hudson (2004) draws on in the development of his mutual dependency account of D+N can all be addressed, thereby obviating the need to assume mutual dependency. Where these arguments are inconclusive, there is no need to adopt mutual dependency. Where I have offered alternative analyses, these are arguments against N as head, in which case mutual dependency should not be adopted. In some cases, I have relied on arguments that D is a quantifier and that it quantifies over N: such an analysis is inherently compatible with the D as head analysis. The advantage of the present account is that it leaves the syntax of D+N asymmetrical and single headed and therefore consistent with the most essential premise of a dependency grammar. It is also more consistent with a word-by-word theory of dependency parsing, and does not involve a learnability problem by introducing a contradiction. I have, however, left open for future research the other mutual dependency structures that Hudson (1990) introduces.

Acknowledgements

I am grateful to Dick Hudson for comments on the first draft; Doug Arnold, Ronnie Cann, Adam Przepiórkowski, Geoff Pullum and Rob Truswell for discussing points of detail (they bear no responsibility at all for what I have done with their help); and three referees for very useful comments which have improved the paper (with apologies that space limitations prevented me from incorporating all of their suggestions). I am, of course, responsible for remaining errors and infelicities.

References

John M Anderson. 2006. *Modern Grammars of Case*. Oxford University Press, Oxford.

Miguel Ballesteros and Joakim Nivre. 2013. Going to the Roots of Dependency Parsing. *Computational Linguistics*, 39(1):5–13, March.

Morten H. Christiansen and Nick Chater. 2016. The Now-or-Never bottleneck: A fundamental constraint on language. *Behavioral and Brain Sciences*, 39:e62.

Allan M Collins and Elizabeth Loftus. 1975. A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407–428.

Jin Cong and Haitao Liu. 2014. Approaching human language with complex networks. *Physics of Life Reviews*, 11(4):598–618, December.

Michael Covington. 2001. A Fundamental Algorithm for Dependency Parsing. In John A. Miller and Jeffrey W. Smith, editors, *Proceedings of the 39th Annual ACM Southeast Conference*, pages 95–102. Association for Computing Machinery.

Chet Creider and Richard Hudson. 2006. Case agreement in Ancient Greek: Implications for a theory of covert elements. In Kensei Sugayama and Richard Hudson, editors, *Word Grammar: New Perspectives on a Theory of Language Structure*, pages 33–53. Continuum, London.

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, November.

Ramon Ferrer i Cancho, Ricard V. Solé, and Reinhard Köhler. 2004. Patterns in syntactic dependency networks. *Physical Review E*, 69(5):051915, May.

Richard Futrell, Roger P. Levy, and Edward Gibson. 2020. Dependency locality as an explanatory principle for word order. *Language*, 96(2):371–412.

Richard Landy Jones Futrell. 2017. *Memory and Locality in Natural Language*. Ph.D. thesis, MIT, Cambridge MA.

Haim Gaifman. 1965. Dependency systems and phrase-structure systems. *Information and Control*, 8:304–337.

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

Daniel Gildea and David Temperley. 2010. Do Grammars Minimize Dependency Length? *Cognitive Science*, 34(2):286–310, March.

Nikolas Gisborne. 2012. The semantics of definite expressions and the grammaticalization of THE. *Studies in Language*, 36(3):603–644.

Nikolas Gisborne. 2020. *Ten Lectures on Events in a Network Theory of Language*. Brill, Leiden.

David G. Hays. 1964. Dependency theory: A formalism and some observations. *Language*, 40(4):511–525.

Rodney Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge.

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

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

Richard Hudson. 2000. I amn't. *Language*, pages 297–323.

Richard Hudson. 2004. Are determiners heads? *Functions of Language*, 11(1):7–42.

Richard Hudson. 2007. *Language Networks: Towards a New Word Grammar*. Oxford University Press, Oxford.

Richard Hudson. 2010. *An Introduction to Word Grammar*. Cambridge University Press, Cambridge.

Timo Järvinen and Pasi Tapanainen. 1998. Towards an implementable dependency grammar. In *Processing of Dependency-Based Grammars*.

Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of Life Reviews*, 21:171–193, July.

Haitao Liu. 2018. Language as a human-driven complex adaptive system. *Physics of Life Reviews*, 26–27:149–151, November.

Ryan McDonald and Joakim Nivre. 2011. Analyzing and Integrating Dependency Parsers. *Computational Linguistics*, 37(1):197–230, March.

Igor A. Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. SUNY press, Albany, New York.

Igor A. Mel'čuk. 2014. Dependency in language. In Kim Gerdes, Eva Hajičová, and Leo Wanner, editors, *Dependency Linguistics: Recent Advances in Linguistic Theory Using Dependency Structures*, pages 1–32. Benjamins, Amsterdam/Philadelphia.

Stephen Neale. 1990. *Descriptions*. MIT Press, Cambridge MA.

Geoffrey Nunberg, Ivan Sag, and Thomas Wasow. 1994. Idioms. *Language*, 70:491–538.

Timothy Osborne. 2003. *The Third Dimension: A Dependency Grammar Theory of Coordination for English and German*. Ph.D. thesis, Pennsylvania State University, State College PA.

Timothy Osborne. 2019. *A Dependency Grammar of English: An Introduction and Beyond*. Benjamins, Amsterdam.

Timothy Osborne. 2021. NPs, not DPs: The NP vs. DP debate in the context of dependency grammar. *Acta Linguistica Academica*, 68(3):274–317.

Barbara Partee. 1975. Montague Grammar and Transformational Grammar. *Linguistic Inquiry*, 6(2):203–300.

Julia Philippi. 1997. The rise of the article in Germanic languages. In Ans Van Kemenade and Nigel Vincent, editors, *Parameters of Morphosyntactic Change*, pages 62–93. Cambridge University Press, Cambridge.

Jane J. Robinson. 1970. Dependency structure and transformational rules. *Language*, 46(2):259–285.

Andrew Rosta. 2006. Structural and distributional heads. In Kensei Sugayama and Richard Hudson, editors, *Word Grammar: New Perspectives on a Theory of Language Structure*, pages 171–203. Continuum, London.

Bertrand Russell. 1905. On denoting. *Mind*, 14(56):479–493.

Uli Sauerland. 1998. *The Meaning of Chains*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Petr Sgall, Eva Hajicová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Reidel, Dordrecht.

Mark Steyvers and Joshua B. Tenenbaum. 2005. The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. *Cognitive Science*, 29(1):41–78, January.

Lucien Tesnière. 1953. *Esquisse d'une Syntaxe Structurale*. Librarie C. Klinksieck, Paris.

Lucien Tesnière. 1959. *Eléments de Syntaxe Structurale*. Librarie C. Klincksieck, Paris.

David S Touretzky. 1986. *The Mathematics of Inheritance Systems*, volume 8. Morgan Kaufmann, Los Altos, CA.

Willy Van Langendonck. 1994. Determiners as Heads? *Cognitive Linguistics*, 5(3):243–259.

A monarchy without subjects: on Brassai's (almost) subject-free dependency grammar

András Imrényi
Eszterházy Károly Catholic University
Eger, Hungary
imrenyi.andras@uni-eszterhazy.hu

Abstract

The paper presents Sámuel Brassai's reasons for almost entirely eliminating the term *subject* (Hu. *alany*) from syntactic analysis, and the manner in which this was achieved. It is shown that Brassai's dependency grammatical theory was in large measure motivated by his rejection (reminiscent of Tesnière) of the logical tradition working with a subject-predicate division. In contrast with much of today's dependency grammar, Brassai did more than relegate subjects to dependent status; he also stripped them of their name, preferring to use the term *nominative* (Hu. *nevező*) instead. The term *subject* was retained for only a subset of finite clauses, and applied on a semantic basis in partial independence from nominative case. The final part of the paper discusses Brassai's approach to the semantics of nominative dependents.

1 Introduction

If today's dependency grammarians were asked to name a few basic types of dependency relations, *subject* would probably quickly spring to their minds. Although the idea of a privileged subject-predicate relationship at the top of the clausal hierarchy has long been discarded in the tradition that DG linguists belong to, the term *subject* itself has survived, perhaps largely owing to another opposition it participates in, namely that of *subject* vs *object*. In the DG community, and especially among linguists working on nominative-accusative languages, a syntactic model eschewing the notion of subjects may seem almost unthinkable.

The goal of this paper is to present just such a model, which also happens to be one of the first completely dependency grammatical theories of syntax to be produced in Europe. It will be shown that when Sámuel Brassai, a Transylvanian polymath of the 19th century, developed his DG approach to the sentence, he did so by stripping subjects not only of their status (as standing in a privileged relationship with the predicate) but also of their name. While no doubt controversial, Brassai's argument deserves close scrutiny, and this is what the present paper aims to accomplish.

The paper is structured as follows. In Section 2, it is discussed why Brassai rejected the dualistic, logically inspired tradition based on the subject-predicate opposition, developing a verb-centric dependency grammatical analysis instead. Section 3 is devoted to Brassai's terminological choice of referring to the relevant dependents as nominatives rather than subjects, and the use that Brassai still found for the latter term. Section 4 outlines Brassai's approach to the semantics of nominatives, which is consonant with recent work in construction grammar. Finally, Section 5 concludes the paper.

2 From dualism to monarchy: how Brassai stripped subjects of their status

Looking for a suitable way for framing the history of dependency grammar, Sériot (2020) remarks that "researchers without any link between each other can reach conclusions that are identical or very similar", one possible reason being that "they reject the same thesis, because they find it unsatisfactory" (Sériot, 2020: 254). The history of DG lends considerable support to Sériot's approach, especially with regard to the idea of verb-centricity. In particular, as the present section aims to show, Brassai's decision to treat the verb as the unique root node of the clause was at least in part motivated by his strong dissatisfaction with a logical tradition that had regarded the subject and the predicate as equally prominent. In this respect, Brassai's position was exactly the same as Tesnière's several decades later, even though there is no reason to assume that Tesnière ever heard about Brassai.

Let us begin with a brief overview of what it is that Brassai and Tesnière would both come to reject. The assumption that subject and predicate are of equal prominence, mutually presupposing each other, has been widely held in linguistics. Inspired by a logical tradition going back to Aristotle, several syntacticians have assumed that sentences have two equally indispensable parts: a subject, expressing that about which something is said, and a predicate, which is what is said about the subject. This view did not only leave its mark on 20th c. constituency-oriented theories (see the S → NP VP rewrite rule in Chomsky, 1957) but had also been present in otherwise dependency-oriented approaches of previous eras. A remarkable example is the following diagram produced by Billroth (1832: 102); for discussion, see Osborne (2020: 191). In the diagram, the subject *Miltiades* and the predicate *reddidit* 'gave back' are both at the top in an otherwise fully DG-compatible analysis of the Latin sentence *Miltiades, dux Atheniensium, toti Graeciae libertatem paene oppressam in pugna apud Marathonem reddidit* 'Miltiades, leader of the Athenians, returned severely oppressed freedom to all of Greece in the battle at Marathon.'

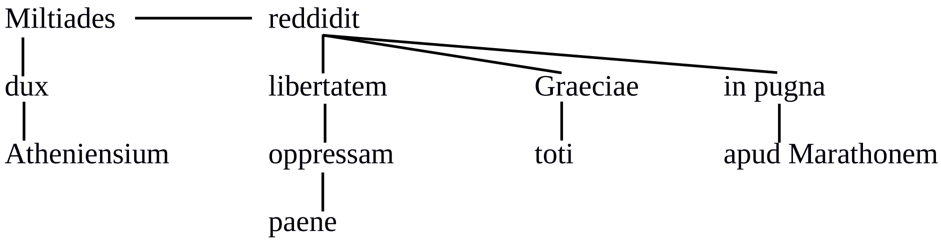


Figure 1: Billroth's (1832) analysis of a Latin sentence.

Whether or not Brassai had access to Billroth's grammar is uncertain but he did know about the Port-Royal analysis and regarded it as a modern version of the dualistic treatment of the sentence in Antiquity (Brassai, 1873: 5; Brassai, 1874: 64, cf. Imrényi and Vladár, 2020: 168). His reasons for refusing to begin syntactic analysis with an initial subject-predicate division are stated in the passage below.[1]

"Therefore the first question is: what are the main parts of the sentence in its topmost division? […] Pressing reasons force me to discard the presumptuous answer: subject and predicate. I will say it briefly: 1.) linguistics being an inductive discipline, I cannot set forth with a principle based on insufficient induction; 2.) in fact, dualism in the sentence is strictly speaking not even the offspring of linguistic induction; rather, it has been borrowed from another discipline, logic; 3.) I am yet to find a linguistic interpretation of subject by which it could be identified in every sentence; 4.) in its implications, the entire principle is unfruitful for the ensuing discussion [in this treatise]; 5.) granted that there might be languages in which it could be ubiquitously employed (even if by force), in Hungarian this is entirely impossible" (Brassai, 2011/1860: 104).

It is interesting to compare Brassai's passage with Tesnière's observations below; it is clear that the two linguists are completely on the same platform.

"Founded on the principles of logic, traditional grammar strives to find the logical opposition between subject and predicate in the sentence, the subject being that about which something is said and

[1] Throughout the paper, Brassai's passages are quoted in my translation (A.I.).

the predicate being what is said about it. Hence in the sentence *Alfred speaks slowly*, the subject would be *Alfred*, and the predicate *speaks slowly* […]. One can acknowledge that this conception of the sentence is merely a remnant that has not yet been entirely eliminated. This remnant stems from the epoch that extends from Aristotle to Port-Royal, when all grammar was founded on logic. Indeed, all arguments that can be invoked against the concept of the verbal node and in favor of the opposition between subject and predicate come a priori from formal logic, which has nothing to do with linguistics" (Tesnière 2015/1966: 98).

As can be seen, both Brassai and Tesnière consider the dualistic analysis to be grounded in logic rather than linguistics, and are critical of its deductive (aprioristic) rather than inductive nature. In what follows, let us examine Brassai's third and fifth arguments in somewhat more detail.

After a survey of conventional meanings of *alany* 'subject' in Hungarian, Brassai notes that linguistics has taken over the meaning this word has in logic, i.e. 'in a logical proposition, the concept whose attribute is specified, or an inferior (more specific) concept which is subsumed by a superior (more general) one'. With some adjustment, grammarians have developed the meaning 'that about which the predicate says something', with the predicate expressing 'that which is said about the subject' (Brassai 2011/1860: 45).

Brassai goes on to challenge this approach by listing sentences in which the above definition can easily lead to an incorrect identification of the subject. Let us now consider only the example below.

(1) Közös lónak túros a háta.
shared horse.DAT worn the back.PX(3SG).NOM
'A shared horse has a worn back.'

The Hungarian proverb in (1) expresses a proposition about shared horses; namely, that their backs are worn. However, grammatically speaking, *lónak* is in dative case. In practice, the nominative *háta* 'back.PX(3SG).NOM' should be identified as subject under the assumptions of traditional grammar. The problem, of course, is that this analysis is hardly backed up by the above logical definition.

The example also lends support to Brassai's fifth argument quoted in the passage above: "granted that there might be languages in which [the logical division of sentences into subject and predicate] could be ubiquitously employed (even if by force), in Hungarian this is entirely impossible" (Brassai, 2011/1860: 104). Further evidence for this comes from the fact that Hungarian weather verbs can act as full-fledged sentences by themselves (e.g. *esik* 'it is raining', *havazik* 'it is snowing', see also It. *piove* 'it is raining'), without any expletive subject. As Brassai notes,

"It is true that Germans cannot say *regnet* by itself but rather need to put a subject-gapfiller *es* before it: *es regnet*, and one cannot blaim them for starting off with the nature of their language and granting such importance to syntactic dualism. But why would a Hungarian adopt their train of thought and resulting bias? To rise to the 'level of science'?" (Brassai 2011/1860: 106).

Brassai thus concludes that the verb alone is at the top of syntactic hierarchy. Just like Mel'čuk (1988: 23) more than a hundred years later, he considers it fundamental that one-word sentences consisting only of a finite verb are a common phenomenon. As Brassai puts it,

"[The verb] can perform the function of the sentence in and by itself, without its apprentices, while these latter cannot possibly exist without their master. *Esik* ['it is raining'], *havazik* ['it is snowing'], *villámlik* ['it is lightning'], *dörög* ['it is thundering'], *kiabálnak* ['they are shouting'], *muzsikálnak* ['they are playing music'], *egyél* ['eat!'], *szaladj* ['run!'], etc. fully express in themselves what the speaker wants to convey. And the hearer need not supplement it or replace it by something else, but comes in immediate and complete possession of the concept that the speaker wished to evoke in him. When someone tells me: *esik*, the whole phenomenon of rain, the darkening of the sky, the fall of raindrops, the dampening of the ground appear in my imagination so fully, even unseen, that the poetic description of a Vörösmarty or Arany [Hungarian poets] could not do better. In this word: *kiabálnak*, the gasping of mouths, the air and the resulting vibration in the hearer's nerves, the sound itself, are all included, thus the event, the subject and the object are fused into a single word to evoke the desired image" (Brassai, 2011/1863: 104–105).

For Brassai, the implications are clear: the finite verb alone is the "soul" of the sentence (Brassai, 2011/1860: 104). Using another metaphor, he also describes it as the monarch of the sentence, a view reminiscent of Dmitrievsky's proposal whereby "the verb is the absolute ruler, the Tsar of the proposition" (Dmitrievsky, 1877: 23, quoted by Sériot, 2020: 264). To quote Brassai's elaborate discussion of the metaphor,

> "Sitting at the beginning, middle, or end of the sentence, wherever it pleases him, is the monarch, the verb, related by meaningful bonds to his vassals, the dependents [*igehatárzók*]. […] The rule of the verb is no dictatorship, and his vassals are no slaves but have lawful relations to their lord and to one another; they each possess a degree of autonomy and a certain rank, with a feudalism whose slogan is, just as in history, *nulle terre sans seigneur* [no land without a lord]" (Brassai, 2011/1860: 48).

With this metaphor, Brassai arrived at nothing less than a complete, coherent dependency grammatical conception of sentences, as argued by Imrényi (2013) and Imrényi and Vladár (2020). It is not the purpose of this paper, however, to repeat the same points that have been made elsewhere. In this section, my aim has been to show that Brassai's view of the sentence as a monarchy grew out naturally from his dissatisfaction with the dualistic logical tradition that had regarded the subject and the predicate as equally prominent. Brassai stripped subjects of their privileged status, and relegated them to the rank of dependents. And this was not all: as shown in the section below, Brassai's conceptual shift also had terminological consequences.

3 Terminological choices: how Brassai stripped subjects of their name

On 4 June 1860, Brassai discussed the problems that examples like (1) posed to the logical definition of subjecthood at an assembly of the Hungarian Academy of Sciences, reading out the first part of his treatise on Hungarian sentences. He fully expected objections to his ideas and often anticipated and reacted to possible counter-arguments that he thought could be coming from the audience. One example for this is the passage below:

> "Not a word, gentlemen! I know the objection at the tip of your tongues. I mean, that I did not have a good grasp of the matter, i.e. 'I did not add to the interpretation of subject the highly characteristic feature that the subject must also be the nominative'! But to this objection let me just say that 'it would have been better for it not to have been born.' Because as soon as we are opening up its secrets, it becomes Pandora's box for you. For one thing, you acknowledge by it that the previously endorsed interpretation is not good, as it needs to be 'supplemented.' Moreover, since the pertinent noun's nominative character appears to determine sufficiently that *it* is the subject, that controversial interpretation is rendered superfluous" (Brassai, 2011/1860: 46).

This argument is probably the key to Brassai's choice of almost entirely eliminating the term *subject* from syntactic analysis, and his preference for the term *nominative*. However, a purely formal description would not satisfy Brassai; he was also interested in the semantic basis of selecting the noun which must be the nominative in a given sentence. The underlying reason was that he placed emphasis not only on sentence analysis but also on synthesis (Brassai, 2011/1860: 47). Therefore, as he put it, "we need to search for a real interpretation, i.e. one based on the nominative's meaning, derived from its relation to the verb" (Brassai, 2011/1864: 200). In Section 4, I return to the issue of how Brassai assigned a semantic interpretation to nominatives. For now, let us continue with the question as to what place, if any, the term *subject* had in Brassai's system.

Brassai did retain the word *subject*, but applied it only to sentences that expressed logical propositions. In his view, this was only the case when a concept's attribute was specified, or an inferior (more specific) concept was subsumed by a superior (more general) one (Brassai, 2011/1860: 45). Two such examples are given below.

(2) A gyermek játszik. (Brassai, 2011/1864: 205)
the child.NOM plays
'The child plays.'

(3) Én pap vagyok. (Brassai, 2011/1864: 239)
I priest.NOM am
'I am a priest.'

With regard to (2), Brassai hastens to note that it is only a logical proposition when it means ’The child is a playing entity/creature’ (or ’Children are playing creatures’), not when it means ’The child is playing (right now)’. He brings up the analogy of a sentence expressing that a leaf is flying in the wind, which does not imply that leaves are flying entities (Brassai, 2011/1864: 206). In (3), which contains two nominative elements, one (*én* ’I’) is analysed as subject and the other (*pap* ’priest’) as attribute. According to Brassai,

“In relation to the attribute, and only that – not in relation to the ‘predicate’ of our grammarians, which also subsumes the verb – can the concept of *subject* come into play. Here we can safely settle for the interpretation according to which it is a noun or pronoun about which the attribute is stated. Conversely, the attribute is a noun, pronoun or adjective that is stated about the subject. *Here*, I say, because in order to get the wind out of a possible objection and to dismiss any accusation of inconsistency, I declare repeatedly that for a theory of the sentence, I consider the concept of subject to be generally barren and useless, no matter how it is interpreted; but in relation to the attribute it has both appropriate meaning and sufficient usefulness” (Brassai, 2011/1864: 242).

To make sense of Brassai’s proposal, it seems necessary to recognize that his restricted notion of subjecthood is semantic rather than grammatical; it concerns meaning in partial independence from formal properties such as nominative case. For example, in the sentence below, Brassai considers the accusative noun *Zsugorit* to function as subject with respect to the attribute *fösvénynek.*

(4) Én Zsugorit fösvénynek tartom. (Brassai, 2011/1864: 242)
I Zsugori.ACC mean.DAT consider.1SG
’I consider Zsugori to be mean.’

To conclude this section, Brassai eliminated the term *subject* from syntactic description, at least as far as its traditional application was concerned. For what other grammarians had called subject, he preferred the term *nominative* (Hu. *nevező*). He reserved the term *subject* (Hu. *alany*) as a semantic category that he applied in the context of *subject-attribute* rather than *subject-predicate* relations, in partial independence from nominative case. What remains to be seen is how Brassai defined the meaning of nominatives; a quest made necessary by his emphasis on sentence synthesis (Brassai, 2011/1860: 47).

4 Brassai’s approach to the semantics of nominative dependents

As hinted above, Brassai was critical of purely formal definitions of grammatical categories on the grounds that he considered them unfit for the purpose of accounting for sentence synthesis (production). Discussing the observation that in English, the nominative (the so-called subject) could be identified by its position in front of the verb, he noted that “in the analysis of a correct and complete sentence, this is definitive indeed; but in synthesis, when I need to produce a sentence in a given language, does it help? Not one bit, because I should figure it out from the meaning of words to be included in the sentence, from their relations to the verb, which one I should or could put in front as subject or nominative” (Brassai, 2011/1860: 47). In the present section, I turn to the question of how Brassai sought to associate a semantic characterization with nominative dependents. As we shall see, he adopted a piecemeal approach relativized to particular constructions (active, passive, middle), in a way that is consonant with recent work in construction grammar. Interestingly, by 1864 he came to assume a position that he had rejected in his 1860 lecture.

Surveying various definitions of subjecthood in his 1860 lecture, Brassai remarks that according to certain linguists, the nominative expresses some (often metaphorical and imaginary) agent or actor. However, since this definition fails to account for all instances of nominative nouns, the definition has had to be extended so that

“the subject is the entity that acts when it is beside an active verb […]; suffers when it is beside a passive verb […]; and beside a neutral verb, it is in a state or involved in the event that is expressed by the verb. This way, it is ‘defined’ indeed, that is true, but it is overly specified. […] Because of its numerous definitions, subject has become a protean concept, so protean that it is extremely hard to grab and downright impossible to comprehend” (Brassai, 2011/1860: 46).

Returning to the problem of semantic characterization in his 1864 lecture, now consistently adopting the term *nominative* rather than *subject*, Brassai seems ready to accept this piecemeal approach, even though he is less than fully satisfied. He proposes the following definition:

"the thing denoted by the nominative is the actor in the plot of active verbs, the sufferer in that of passive verbs, and it is in a particular state in the plot of middle verbs. The generalization cannot be taken any further, hence the true [semantic] interpretation cannot be considered completely successful" (Brassai, 2011/1864: 201).

From the perspective of present-day construction grammar, Brassai had no reason to be dissatisfied. What he did was develop a set of construction-specific definitions of the meaning(s) of nominative dependents. Rather than seeking to define the meaning of nominative dependents as such, he settled for defining the meaning of nominative dependents of transitive verbs, passive verbs, middle verbs, etc. (cf. Comrie, 1978; Dixon, 1979; Croft, 2001: 134). Theoretically, this approach is justified at length by Croft (2001), who argues that "constructions, not categories and relations, are the basic, primitive units of syntactic representation", and consequently that "[t]he categories and relations found in constructions are derivative" (Croft, 2001: 46). In terms of language acquisition, the point that form-meaning correspondences are always learnt in particular contexts, and are conditioned by those contexts, is convincingly made by Ellis (2006):

"Learners FIGURE language out: their task is, in essence, to learn the probability distribution P (interpretation|cue, context), the probability of an interpretation given a formal cue in a particular context, a mapping from form to meaning conditioned by context" (Ellis, 2006: 8, quoted by Gries, 2017: 593).

To conclude this section, Brassai's approach to sentence structure may be seen as falling into the traditions of both dependency grammar and construction grammar (or more broadly, cognitive linguistics).[2] Not only did he propose a verb-centric, dependency-based description of clause structure but he also insisted on the study of form-meaning correspondences rather than accepting purely formal definitions of grammatical categories. In fact, his work seems to have been guided by a principle that takes the following form in Langacker's Cognitive Grammar: "all constructs validly posited for grammatical description (e.g. notions like "noun", "subject", or "past participle") must in some way be meaningful" (Langacker, 2008: 5).

5 Summary and conclusions

The goal of the paper was to observe the fate of subjects in Sámuel Brassai's dependency grammatical theory of the sentence. As noted in Section 2, Brassai's concept of a verb-centric, monarchy-like structure in the sentence grew out naturally from his dissatisfaction with the logical tradition that worked with an initial subject-predicate division. In Section 3, I discussed Brassai's reasons for preferring the term *nominative* to *subject*; in short, he was not content with any of the existing definitions of subjecthood, and considered the term *nominative* to provide a better basis. He retained the term *subject* in the context of *subject-attribute* rather than *subject-predicate* relations, and employed it in a semantic sense in partial independence from nominative case. Finally, Section 4 addressed the question as to how nominative dependents could receive a semantic characterization. Here, Brassai (somewhat unwillingly) endorsed a piecemeal approach, relativized to particular constructions. Specifically, he argued that "the thing denoted by the nominative is the actor in the plot of active verbs, the sufferer in that of passive verbs, and it is in a particular state in the plot of middle verbs" (Brassai, 2011/1864: 201).

Brassai's ideas are not only of historical interest; rather, they may inform theory development in present-day linguistics. Among others, the following points seem to be worthy of serious consideration by those working in dependency grammar:

1. The idea that the notion of subject should not be taken for granted in DG. It is a remnant of a logical tradition (cf. Tesnière 2015/1966: 98), and simply relegating subjects to dependent status may not be enough for completely eliminating that tradition's potentially undesirable implications.

[2] For suggestions that the tenets of dependency grammar and construction grammar can be combined, see e.g. Hudson (2008), Welke (2011), Osborne and Gross (2012) and Imrényi (2017).

2. The idea that purely formal grammatical categories are not satisfactory. In order to account for sentence synthesis, it is necessary to explore the semantic basis of such formal properties as case assignment and word order, with the aim of defining grammatical categories in terms of form-meaning correspondences.
3. The idea that “every language and every construction [should] be characterized in its own terms” (Langacker, 2008: 423, see also Haspelmath, 2015). With regard to syntactic dualism, Brassai passionately argues that what might be a well-motivated generalization in a grammar of German is clearly not optimal for Hungarian (Brassai 2011/1860: 106). In his description of the meaning of nominative dependents, he endorses a set of construction-specific definitions.

All in all, the paper forms part of an attempt aimed at demonstrating that Brassai’s ideas fall within the traditions of both dependency grammar and construction grammar (two schools of thought whose past is much longer than their history), potentially informing their integration as well.

Acknowledgments

The research behind this paper was supported by the Bolyai János Research Fellowship of the Hungarian Academy of Sciences and the ÚNKP-21-5 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund (NKFI). Further support was received from NKFI project K129040.

Innovációs és Technológiai Minisztérium

ÚNKP
Új Nemzeti Kiválóság Program

National Research, Development and Innovation Office
Hungary

References

Johann Gustav Friedrich Billroth. 1832. *Lateinische Syntax für die obern Klassen gelehrter Schulen*. Weidmann, Leipzig.

Sámuel Brassai. 2011/1860. A magyar mondat. I. értekezés. [The Hungarian sentence. First treatise.] In. *A magyar mondat*. Válogatta Elekfi László és Kiefer Ferenc. [The Hungarian sentence. Texts selected by László Elekfi and Ferenc Kiefer.] Tinta, Budapest. 10–95.

Sámuel Brassai. 2011/1863. A magyar mondat. II. értekezés. In. *A magyar mondat*. [The Hungarian sentence. Second treatise.] Válogatta Elekfi László és Kiefer Ferenc. [The Hungarian sentence. Texts selected by László Elekfi and Ferenc Kiefer.] Tinta, Budapest. 99–189.

Sámuel Brassai. 2011/1864. A magyar mondat. III. értekezés. [The Hungarian sentence. Third treatise.] In: *A magyar mondat*. Válogatta Elekfi László és Kiefer Ferenc. [The Hungarian sentence. Texts selected by László Elekfi and Ferenc Kiefer.] Tinta, Budapest. 192–356.

Sámuel Brassai. 1873. *Paraleipomena kai diorthoumena. A mit nem mondtak s a mit roszul mondtak a commentatorok Virg. Aeneise II. könyvére*. [What the commentators did not say or wrongly said about Book II of Virgil’s Aeneid.] MTA, Budapest.

Sámuel Brassai. 1874. *Laelius. Hogyan kell és hogyan nem kell magyarázni az iskolában a latin autorokat?* [On how Latin authors should and should not be interpreted at schools]. Stein, Kolozsvár.

Noam Chomsky. 1957. *Syntactic structures*. Mouton, The Hague.

Bernard Comrie. 1978. Ergativity. In: Lehmann, Winfrid (ed.), *Syntactic typology*. University of Texas Press, Austin. 329–394.

William Croft. 2001. *Radical Construction Grammar*. Oxford University Press, Oxford.

Robert M. W. Dixon. 1979. Ergativity. *Language* 55:59–138.

Alexey Dmitrievsky. 1877. Prakticheskie zametki o russkom sintaksise, II: Dva li glavnykh chlena v predlozhenii? [Practical notes on Russian syntax: Are there two main members of the proposition?]. *Filologicheskie Zapiski* (Voronezh), 4:15–37.

Nick C. Ellis. 2006. Language acquisition as rational contingency learning. *Applied Linguistics,* 27(1):1–24.

Stefan Th. Gries. 2017. Corpus approaches. In: Dancygier, Barbara (ed.), *The Cambridge Handbook of Cognitive Linguistics.* Cambridge University Press, Cambridge. 590–606.

Martin Haspelmath. 2015. Framework-free grammatical theory. In: Heine, Bernd and Narrog, Heiko (eds.), *The Oxford handbook of grammatical analysis*. 2nd edition. Oxford University Press, Oxford. 287–310.

András Imrényi. 2013. Constituency or dependency? Notes on Sámuel Brassai's syntactic model of Hungarian. In Szigetvári, Péter (ed.), *VLlxx. Papers presented to László Varga on his 70th birthday*. Tinta, Budapest. 167–182.

András Imrényi. 2017. Form-meaning correspondences in multiple dimensions: The structure of Hungarian finite clauses. *Cognitive Linguistics,* 28(2):287–319.

András Imrényi and Zsuzsa Vladár. 2020. Sámuel Brassai in the history of dependency grammar. In: Imrényi, András and Mazziotta, Nicolas (eds.), *Chapters of Dependency Grammar: A historical survey from Antiquity to Tesnière*. John Benjamins, Amsterdam. 164–187.

Ronald W. Langacker. 2008. *Cognitive Grammar: A basic introduction*. Oxford University Press, Oxford.

Igor Mel'čuk. 1988. *Dependency Syntax: Theory and practice*. State University Press of New York, Albany.

Timothy Osborne and Thomas Gross. 2012. Constructions are catenae: Construction grammar meets dependency grammar. *Cognitive Linguistics*, 23(1):165–216.

Timothy Osborne. 2020. Franz Kern: An early dependency grammarian. In: Imrényi, András and Mazziotta, Nicolas (eds.), *Chapters of Dependency Grammar: A historical survey from Antiquity to Tesnière*. John Benjamins, Amsterdam. 190–213.

Patrick Sériot. 2020. The Russian trail: Dmitrievsky, the little drama metaphor and dependency grammar. In: Imrényi, András and Mazziotta, Nicolas (eds.), *Chapters of Dependency Grammar: A historical survey from Antiquity to Tesnière*. John Benjamins, Amsterdam. 253–275.

Lucien Tesnière. 2015/1966. *Elements of structural syntax.* Translated by Timothy Osborne and Sylvain Kahane. John Benjamins, Amsterdam.

Klaus Welke. 2011. *Valenzgrammatik des Deutschen. Eine Einführung*. De Gruyter, Berlin.

BINGO: A Dependency Grammar Framework to Understand Hardware Specifications Written in English

Rahul Krishnamurthy
Bradley Department of Electrical
and Computer Engineering
Virginia Tech, Blacksburg, VA, USA
rahulk4@vt.edu

Michael S. Hsiao
Bradley Department of Electrical
and Computer Engineering
Virginia Tech, Blacksburg, VA, USA
mhsiao@vt.edu

Abstract

Automatic understanding of specifications containing flexible word order and expressiveness close to natural language is a challenging task. We address this challenge by modeling semantic parsing as a game of BINGO with dependency grammar. In this model, the rows in a BINGO chart of a word represent distinct interpretations, and the columns describe the constraints required to complete each of these interpretations. BINGO parsing considers the context of each word in the input specification to ensure high precision in the creation of semantic frames. We encode contextual information of the hardware verification domain in our grammar by adding semantic links to the existing syntactic links of the link grammar. We also define semantic propagation operations as declarative rules that are executed for each dependency edge of the parse tree to create a semantic frame. We used hardware design specifications written in English to evaluate the framework. Our results showed that the system could translate highly expressive specifications. It also demonstrated the ease of creating rules to generate the same semantic frame for specifications with the same meaning but different word order.

1 Introduction

Automatic understanding of natural language specification documents for hardware and software has numerous benefits such as reduced verification efforts for debugging the design, detection of incomplete specifications, reduced time to fabricate the chip (Ray et al., 2016), etc. The formal output generated by a semantic parser is non-intuitive, and the user may not be able to validate its correctness unless the output is executed. However, natural language specifications are generally written at the early design stages when an executable prototype is not yet available. As a result, the semantic parser output cannot be immediately executed and verified. It becomes the responsibility of the parser to generate only correct translations of the natural language specifications. As evident in (Gu et al., 2016; Lin et al., 2018), machine learning-based semantic parsing approaches require thousands of input-output examples to achieve high accuracy. Unavailability of large number of examples resulted in many rule-based translation works like (Dutle et al., 2020; Giannakopoulou et al., 2020; Mavridou et al., 2020). These rule-based approaches achieve high accuracy in understanding specifications by imposing strict restrictions on the order of words in the input specifications.

This paper presents a dependency grammar-based framework to understand hardware specifications written in English. The grammar is not as rigid as the grammars in the existing works and allows flexibility in the word order variations and input sentence structures. Our framework comprises of two distinct components. The first component is a declarative specification of rules analogous to creating BINGO charts for each word. The second component is a chart parser that takes the BINGO chart of each word as input and performs two steps similar to the game of BINGO: The first step marks cells in the chart of each word. The second step selects a single horizontal BINGO row that passes through the rows of charts of all the words and covers only marked cells.

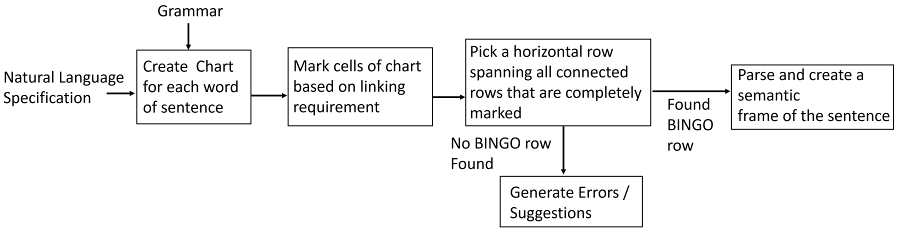

Grammar
Natural Language Specification
Create Chart for each word of sentence
Mark cells of chart based on linking requirement
Pick a horizontal row spanning all connected rows that are completely marked
Found BINGO row
Parse and create a semantic frame of the sentence
No BINGO row Found
Generate Errors / Suggestions

Figure 1: Framework for parsing and translating hardware specification to Semantic Frames.

The declarative rules component is inspired by the syntactic link rules of the link grammar. In our grammar, we have added semantic links to the existing syntactic links of the link grammar. The semantic links serve the following purposes: (1) they represent the semantic context with which a word can be used in the specification, (2) they define semantic propagation operations that should be executed when a link between two words is created.

We build semantic frames for a hardware specification compositionally using the link parse tree as the syntactic structure. The creation and propagation of the semantic frames from the individual nodes to the root node of the parse tree are governed by the semantic propagation rules defined in the semantic links.

Figure 1 shows the flow of the major modules of our framework. We first create a BINGO chart for each word in a given specification based on the underlying grammar. The BINGO parser marks cells in the BINGO chart of each word according to the word's syntactic and semantic links. After completing the marking process, we search for a BINGO row that is a horizontal row spanning the marked rows of all the charts. The BINGO row represents a solution to connect all the words in the sentence after taking into account each word's context. In order to create a semantic frame, we execute semantic propagation rules for each connection in the BINGO row in a transition-based dependency parsing framework.

The rest of the paper is organized as follows. Section 2 discusses the previous works that have employed dependency structures to understand specifications. To better explain the working of our framework, we introduce the data structure used in our framework in section 3. In section 4, we present different components of grammar. Section 5 covers the parsing methodology of the framework. In section 6, we discuss the evaluation of our work. Finally, a concluding summary with the future work is described in section 7.

2 Related Work

Dependency parse trees are useful in extracting semantic relations between entities and have close correspondence to the semantic representation of the sentence (De Marneffe and Nivre, 2019; Covington, 2001). Driven by these advantages, dependency parsing has been used as the core component in the recent works (Ghosh et al., 2016; Yan et al., 2015; Soeken et al., 2014; Nan et al., 2021; Zhang et al., 2020; Chhabra et al., 2018) to automatically understand input specifications written in natural language. However, due to the lack of domain-specific data to train the dependency parser, an off-the-shelf dependency parser trained on general natural language is employed in these applications. It has been shown in literature (Gildea, 2001) that the accuracy of syntactic parser may reduce when applied on text outside its training corpus. Work in (Ghosh et al., 2016; Zhang et al., 2020; Chhabra et al., 2018) attempts to improve the accuracy of the syntactic parser by introducing a pre-processing step to recognize and parse the domain-specific phrases. However, no concrete solution was proposed to resolve ambiguities like preposition and coordination attachments in syntactic parsing. Parse trees with incorrect attachments between words are ineffective for any down-stream natural language understanding application. As pointed out in (Bajwa et al., 2012), the main reason for the inaccuracy in syntactic parsing of specifications is the absence of domain-specific context knowledge and its integration with the parser. More recently, in (Hsiao, 2018), (Hsiao, 2021), domain-specific context knowledge is used to parse specifications for video games, and suggestions and warning messages are given for sentences that could not be handled.

We propose a grammar-based understanding framework that considers context on both the left side and right side of a word before making a dependency arc on the word. Our dependency grammar is

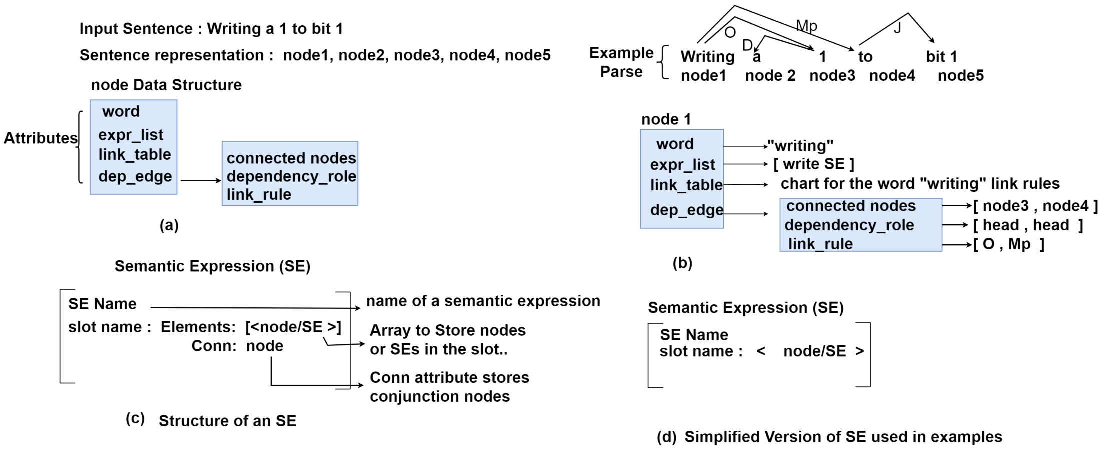


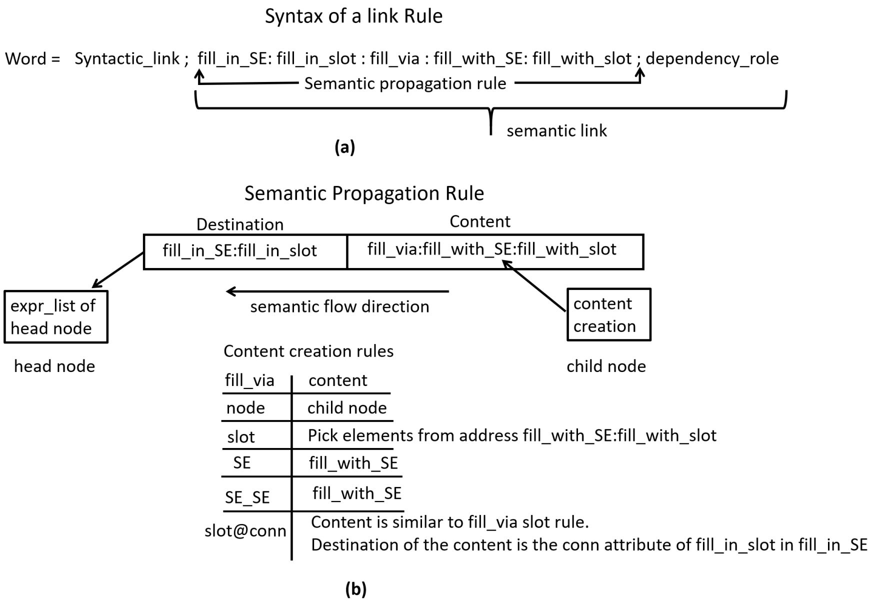
Figure 2: (a) A node data structure to store word and its dependency and semantic information. (b) An example that shows the representation of a node for the word “writing”. (c) A semantic expression (SE) structure to store elements in the slot of an SE. (d) A simplified version of SE without Elements and Conn attributes.

inspired by the link grammar formalism (Sleator and Temperley, 1993) that allows encoding of syntactic context by using binary associative operators ‘&’ and ‘*or*’. The existing link grammar parser is a complicated algorithm that is similar to finding an optimal triangulation of a convex polygon using dynamic programming (Sleator and Temperley, 1993). Also, the link grammar has no provision to define semantics for link rules. We propose a simple parser for the link grammar by modeling link parsing as a game of BINGO. Moreover, we define semantics for each link connection of a word and then combine the semantics of each link connection in a transition-based dependency parsing framework. The final output of the parser is a semantic frame that corresponds to the meaning of the input specification written in the English language. The final semantic frame can be translated to a System Verilog Assertion (SVA) if all the Register Transfer Level (RTL) information is available in the semantic frame.

Earlier work on chart parser in (Nasr and Rambow, 2004) extended a CKY parser of context-free grammar to parse a dependency grammar. The chart items of the CKY parser contained finite state machines, and the chart parsing produced a dependency tree from a packed parse forest in two steps. In the first step, binary syntagmatic trees were extracted from the packed parse forest, and in the second step, each syntagmatic tree was transformed into a dependency tree. In contrast to (Nasr and Rambow, 2004), cells in our chart contained links that connect two words of the input sentence. In our chart parser, a fully connected dependency tree is extracted from the chart in a single step that involved searching for a BINGO row. A BINGO row provided the linkages that connect all the input sentence words in a dependency relation.

3 Data Structure

In our framework, the words of an input specification are represented as nodes of a tree. A node data structure consists of four attributes as shown in Figure 2 (a). The purpose of these attributes are as follows: (1) a word attribute is needed to store the node’s word as a string, (2) expr_list keeps an array of semantic expressions that are either created at the node or propagated to the node in a dependency tree, (3) link_table contains a chart for the node’s link rules, (4) a depedge stores the dependency edge information for the node. Figure 2 (b) illustrates the node representation for the word “writing”. In Figure 2 (c), the structure of the Semantic Expression (SE) is shown for the ease of explaining semantic frames in the subsequent sections. An SE is similar to a semantic frame and has semantic slots. A semantic slot has two attributes Elements and Conn. Elements is an array to store slot values that are either a node or another SE. Conn of the slot contains conjunction nodes like ‘and’ ,‘or’ that connects values of the elements array. In the examples of the subsequent sections, we will use a simplified version of SE shown in Figure 2 (d) that does not explicitly mention Elements and Conn.


Syntax of a link Rule
Word = Syntactic_link ; fill_in_SE: fill_in_slot : fill_via : fill_with_SE: fill_with_slot ; dependency_role
Semantic propagation rule
semantic link
(a)
Semantic Propagation Rule
Destination
Content
fill_in_SE:fill_in_slot
fill_via:fill_with_SE:fill_with_slot
expr_list of head node
head node
semantic flow direction
content creation
child node
Content creation rules
(b)

fill_via	content
node	child node
slot	Pick elements from address fill_with_SE:fill_with_slot
SE	fill_with_SE
SE_SE	fill_with_SE
slot@conn	Content is similar to fill_via slot rule. Destination of the content is the conn attribute of fill_in_slot in fill_in_SE

Figure 3: Structure of (a) a grammar link rule and (b) a semantic propagation rule.

4 Grammar

A link rule in our proposed grammar consists of a syntactic link and a semantic link. Figure 3 (a) shows a general structure of a link rule in the grammar. The syntactic link is taken from the link grammar (Sleator and Temperley, 1993). The meaning of syntactic link rules can be found in (Sleator and Temperley, 1991).

Similar to the notion of syntactic links that represent the linking requirement of a word with its left and right words, we present semantic links that represent the linking requirement for the semantic composition of nodes in a dependency relation. Our semantic links consist of a semantic propagation rule and an indicator of the node's dependency role in the dependency relation.

In the link grammar, two matching syntactic link connectors have the same name and different polarity of + and - directions. Similarly, in our grammar two matching semantic links have the same semantic propagation rule but have an opposite polarity of head and child dependency roles. The semantic composition between two nodes is possible only if they have matching semantic links.

In our framework, we connect two nodes in a dependency relation only if they satisfy the criteria of both syntactic and semantic linking as illustrated in Figure 4. The syntactic link in this figure demonstrates that the node 'writing' expects an object (O+) on its right side, and the (O-) at node '1' indicates that it can be connected as an object of a node on its left side. In this figure, semantic composition is possible between the nodes 'writing' and '1' because they have the same semantic propagation rule write_frame:write_what_value:node where the node 'writing' is the head, and '1' is the child node.

A semantic propagation rule in a semantic link consist of five parameters separated by ":" as shown in figure 3 (a). A semantic propagation rule is needed to perform two tasks. The first task is to define the

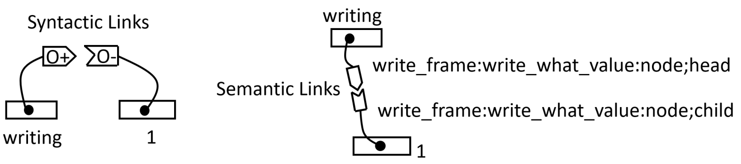
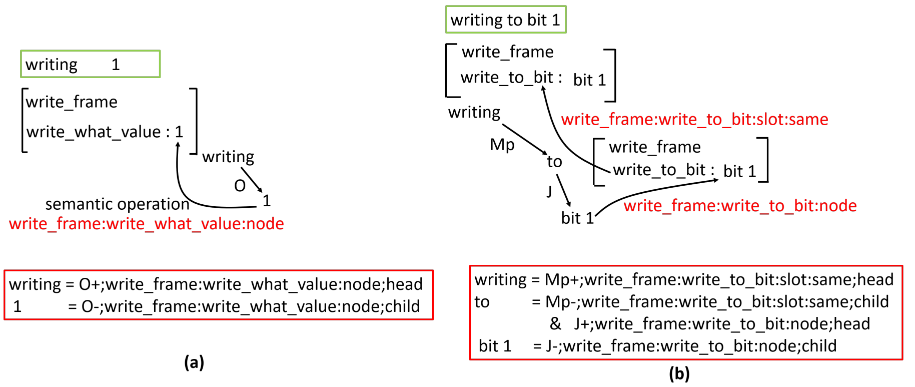

Syntactic Links
O+
O-
writing
1
writing
Semantic Links
write_frame:write_what_value:node;head
write_frame:write_what_value:node;child
1

Figure 4: Syntactic links shows syntactic requirement of words in a sentence. Semantic links represent similar linking requirement between the head and child words in a dependency relation.

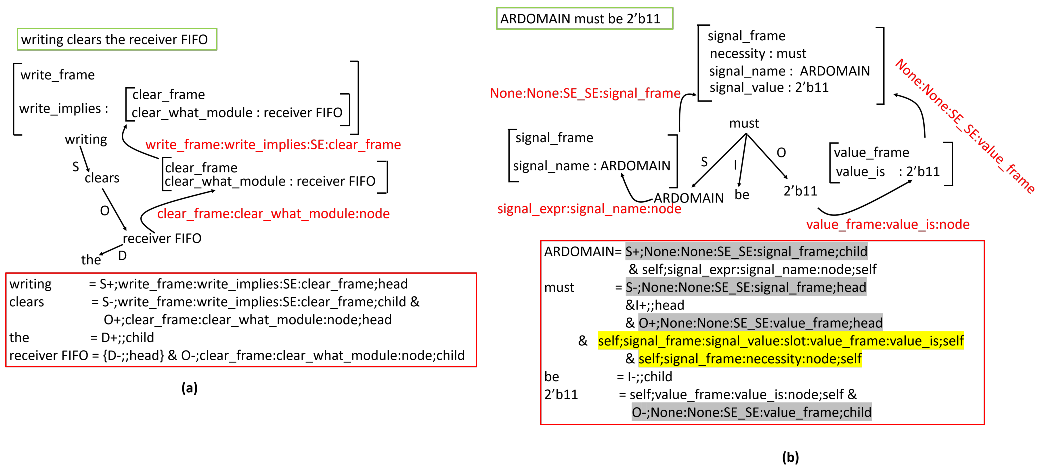


```
writing = O+;write_frame:write_what_value:node;head
1       = O-;write_frame:write_what_value:node;child
```

```
writing = Mp+;write_frame:write_to_bit:slot:same;head
to      = Mp-;write_frame:write_to_bit:slot:same;child
          &  J+;write_frame:write_to_bit:node;head
bit 1   = J-;write_frame:write_to_bit:node;child
```

Figure 5: (a) Example of Semantic propagation operation when fill_via is node. (b) Example of Semantic propagation operation when fill_via is slot.

semantic content that will be propagated from the child node in a dependency relation. The second task is to define the destination SE and slot of the head node where the content will be transferred. Figure 3 (b), illustrates the function of the semantic propagation rule parameters. The first task is accomplished by the following three parameters of the rule: fill_via, fill_with_SE, and fill_with_slot. These parameters create semantic content according to the content creation rules shown in the Figure 3 (b). When the parameter fill_via is node, then the content to be propagated is the child node. Elements of a particular slot (fill_with_slot) from an SE (fill_with_SE) of the child node's expr_list is propagated when the fill_via value is slot. In the case of an SE or SE_SE, we propagate an SE from the expr_list of the child node defined by the fill_with_SE parameter of the rule.


writing clears the receiver FIFO
write_frame
write_implies :
clear_frame
clear_what_module : receiver FIFO
writing
S
clears
write_frame:write_implies:SE:clear_frame
clear_frame
clear_what_module : receiver FIFO
O
clear_frame:clear_what_module:node
receiver FIFO
the
D
(a)
ARDOMAIN must be 2’b11
signal_frame
necessity : must
signal_name : ARDOMAIN
signal_value : 2’b11
None:None:SE_SE:signal_frame
None:None:SE_SE:value_frame
must
signal_frame
signal_name : ARDOMAIN
S
I
O
value_frame
value_is : 2’b11
ARDOMAIN
be
2’b11
signal_expr:signal_name:node
value_frame:value_is:node
(b)

```
writing       = S+;write_frame:write_implies:SE:clear_frame;head
clears        = S-;write_frame:write_implies:SE:clear_frame;child &
                O+;clear_frame:clear_what_module:node;head
the           = D+;;child
receiver FIFO = {D-;;head} & O-;clear_frame:clear_what_module:node;child
```

```
ARDOMAIN= S+;None:None:SE_SE:signal_frame;child
          & self;signal_expr:signal_name:node;self
must     = S-;None:None:SE_SE:signal_frame;head
          &I+;;head
          & O+;None:None:SE_SE:value_frame;head
     &   self;signal_frame:signal_value:slot:value_frame:value_is;self
          & self;signal_frame:necessity:node;self
be       = I-;;child
2’b11    = self;value_frame:value_is:node;self &
           O-;None:None:SE_SE:value_frame;child
```

Figure 6: (a) Example of Semantic propagation operation when fill_via is SE. (b) Example of Semantic propagation operation when fill_via is SE_SE.

The destination of the semantic content is determined by the fill_in_SE and fill_in_slot parameters of the semantic propagation rules. These parameters refer to the SE and slot of the head node that are stored in the expr_list of the head node.

In the fill_via slot propagation rule, the content in the source slot's elements array and conn parameter is propagated to the corresponding elements array and conn parameter of the destination slot in head node's SE. To transfer the content of the elements array from the source slot to the conn parameter of the destination slot, we created a fill_via slot@conn rule. The rule is needed to transfer conjunction nodes like 'and' , 'or' from source slot's elements array to the conn parameter of the destination slot.

We illustrate the working of different semantic propagation operations in Figure 5 and Figure 6. In these figures, we have represented the example phrase in top green box and the corresponding grammar rules in the red box at the bottom of the figure.

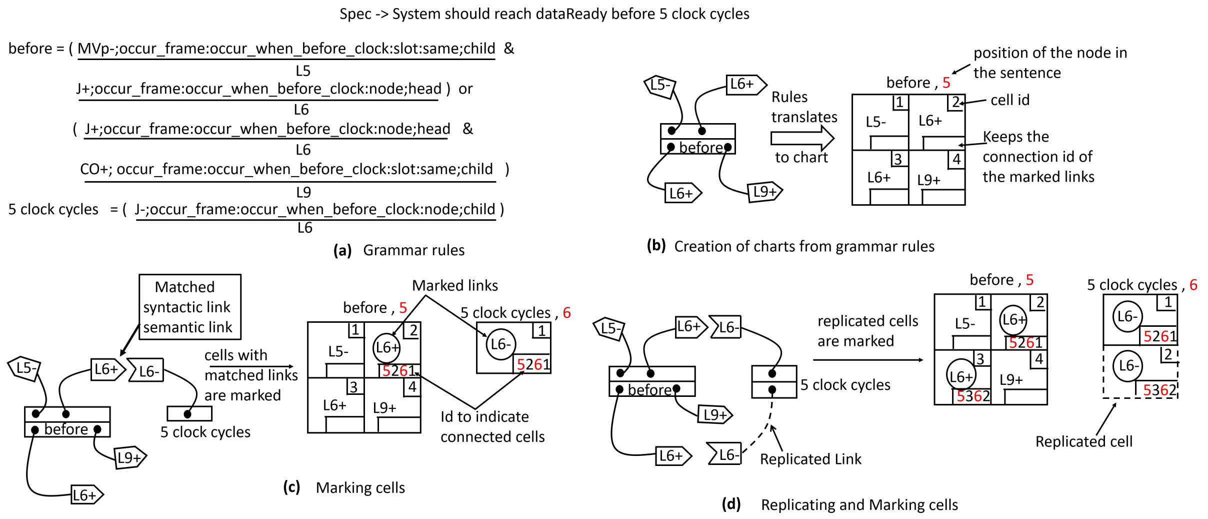

Spec -> System should reach dataReady before 5 clock cycles
before = (MVp-;occur_frame:occur_when_before_clock:slot:same;child &
L5
J+;occur_frame:occur_when_before_clock:node;head) or
L6
(J+;occur_frame:occur_when_before_clock:node;head &
L6
CO+; occur_frame:occur_when_before_clock:slot:same;child)
L9
5 clock cycles = (J-;occur_frame:occur_when_before_clock:node;child)
L6
(a) Grammar rules
L5-
L6+
before
L6+
L9+
Rules translates to chart
before , 5
position of the node in the sentence
cell id
Keeps the connection id of the marked links
L5-
L6+
L6+
L9+
1
2
3
4
(b) Creation of charts from grammar rules
Matched syntactic link semantic link
L5-
L6+
L6-
before
L9+
L6+
5 clock cycles
cells with matched links are marked
Marked links
before , 5
5 clock cycles , 6
L5-
L6+
5261
L6+
L9+
L6-
5261
Id to indicate connected cells
(c) Marking cells
L5-
L6+
L6-
before
L9+
L6+
L6-
5 clock cycles
Replicated Link
replicated cells are marked
before , 5
L5-
L6+
5261
L6+
5362
L9+
5 clock cycles , 6
L6-
5261
L6-
5362
Replicated cell
(d) Replicating and Marking cells

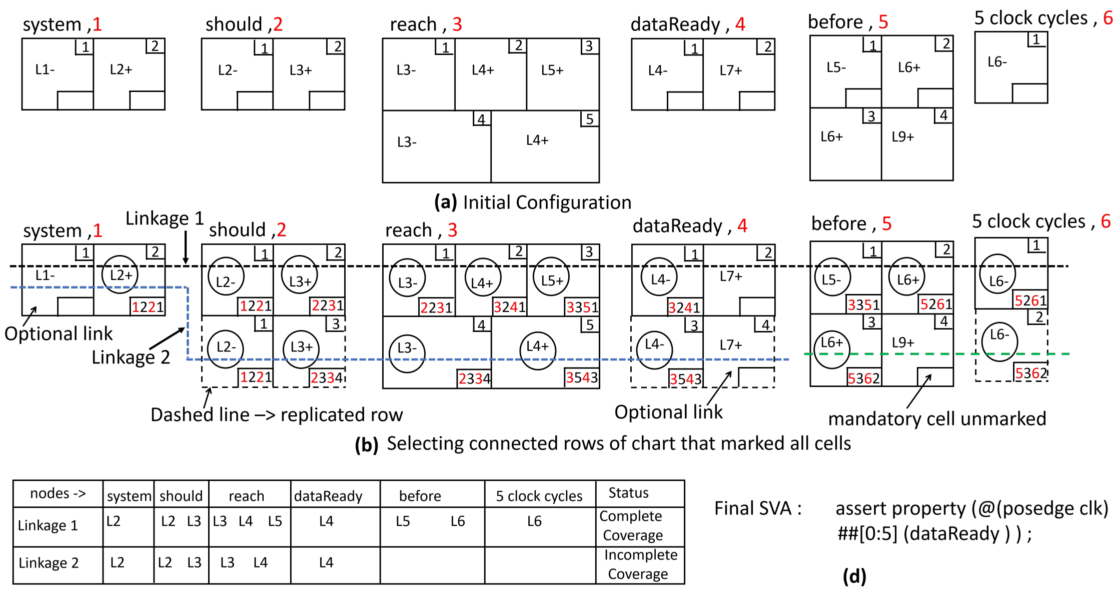
Figure 7: (a) Grammar link rules for words 'before' and '5 clock cycles' to parse ' system should reach dataReady before 5 clock cycles'.(b) Translating grammar rules to BINGO chart. (c) Marking cells of links that have matched syntactic and semantic links without violation link order constraints. (d) Link replication and marking cells of replicated link.

The fill_via node semantic propagation operation represents a scenario where a child node of the dependency edge directly fills the slot of an SE located in the head node's expr_list. The application of this rule between nodes 'writing' and '1' is illustrated in Figure 5 (a). As shown in the grammar rules, the words in dependency relation satisfy each other's syntactic link and semantic link requirements. The semantic fill_via node operation is carried out in dependency parsing by placing the node '1' in the slot write_what_value of the head node's SE write_frame.

A child node of a dependency edge cannot always be a direct argument of the head node's SE. A child node can also act as a bridge propagating semantic information between its connected nodes. We can express a child node as a bridge in our grammar by using fill_via slot and fill_via SE semantic propagation operations. Figure 5 (b) illustrates fill_via slot operation where the node 'to' acts as a bridge. In this figure, the node 'to' plays the role of both head and child node in the conjunctive rule (Mp-;write_frame:write_to_bit:slot:same;child & J+;write_frame:write_to_bit:node;head). The node 'to' receives the content of the slot write_to_bit in the write_frame from the child node 'bit 1' using fill_via node semantic operation. This content is then transferred to the node 'writing' using fill_via slot semantic operation through the edge Mp. The word 'same' in the rule is used for the ease of writing semantic propagation rules that have the same source and destination parameters. For example, in the fill_via slot rule of the word 'to', 'same' indicates that parameters fill_with_SE and fill_with_slot have values equal to the values of the parameters fill_in_SE and fill_in_slot.

Figure 6 (a) illustrates a fill_via SE operation where the node 'clears' creates a clear_frame and passes it to a slot in its head node's SE. In this figure, the SE at the node 'clears' receive 'receiver FIFO' in its slot using fill_via node operation. The entire SE created at the node 'clears' is sent to a slot in the writing node's write_frame using fill_via SE semantic operation. It can be seen in this figure that every syntactic edge is not associated with a semantic propagation operation. In the red box at the bottom of the figure, the rule for syntactic edge 'D' has an empty semantic propagation rule that indicates the absence of a semantic propagation operation when a syntactic edge is created.

Figure 6 (b) shows that the grammar can also express the composition operation of two SE's. This figure illustrates grammar rules and semantic operations for the phrase 'ARDOMIAN must be 2'b11'. The node 'must' receives SE's from its subject ARDOMAIN and object 2'b11 using fill_via SE_SE operation as highlighted in the grey color in the red box. SE_SE semantic operation transfers the entire SE of the child node to its head node's expr_list. In this rule, the SE is not transferred to a slot of an SE in the head node's expr_list. This is defined in the SE_SE rule by specifying 'None' in both the fill_in_SE



(a) Initial Configuration

(b) Selecting connected rows of chart that marked all cells

nodes ->	system	should	reach	dataReady	before	5 clock cycles	Status
Linkage 1	L2	L2 L3	L3 L4 L5	L4	L5 L6	L6	Complete Coverage
Linkage 2	L2	L2 L3	L3 L4	L4			Incomplete Coverage

(c) Output of 2 Step BINGO parsing

Final SVA : assert property (@(posedge clk) ##[0:5] (dataReady)) ;

(d)

Figure 8: (a)Initial config. of Charts based on grammar rules given in Figure 9. (b) Searching BINGO rows after marking cells of charts. (c) Final Linkage solution of the BINGO row shown by the Black dashed line (Linkage 1). (d) System Verilog Assertion (SVA) created from the SE of the parse tree.

and fill_in_slot parameters of the rule. Since the SE's have already reached the 'must' node's expr_list, we combine them using self link rules highlighted in yellow color in the red box in the Figure 6 (b). We created self link rules in the grammar that allows to manipulate SE's that exist at the node's expr_list. The self link rule doesn't represent a connection between nodes and has a dummy syntactic link component (self) that is not matched with the syntactic link of any other node. The word 'self' in the dependency role of the self link rule indicates that the final semantic information created by the self link rule remains at the node and is not propagated to its head node.

5 2-Step BINGO Parser

In parsing, we search for all possible set of links that can connect all the nodes of the input specification in a dependency parse tree and satisfy the syntactic-semantic link requirements of all the nodes. This set of links are called linkages in (Sleator and Temperley, 1991). Figure 8(c) shows the final linkages found at the end of parsing for the input specification 'system should reach dataReady before 5 clock cycles'.

The working of the parser can be visualized as a game of BINGO. The grammar rules of each node can be arranged in a chart where the rows in the chart represent all possible syntactic-semantic links for a specific interpretation of the node. Each cell in a row of a chart contains a link from the node's grammar rule. The total number of cells in a row represents the total number of links required by the node to complete an interpretation. For example, in Figure 7 (a) , the grammar for the node 'before' allows two different combinations of syntactic-semantic connections. This set of rules for 'before' translates to two rows of the chart as shown in the Figure 7 (b). In the chart, each row represents a unique combination of conjunct links that a node can have. For example, the chart for the node 'before' implies that the node 'before' should have either L_5 and L_6 or L_6 and L_9 links connected to it in a dependency parse tree.

Charts of all the nodes are given as input to the parser. The parser performs the following two tasks. First, the parser marks cells of all the charts based on the links inside the cells. Secondly, the parser searches for a set of BINGO rows that are horizontal rows spanning charts of all the nodes and covers only marked cells. A BINGO row represents a linkage that has satisfied all the syntactic-semantic constraints to build the parse tree.

Step 1 Marking Chart cells: Our algorithm marks the cells of the charts as follows: We pick a node and match the cells of its chart with the chart cells of all the previous nodes of the sentence. We mark and connect a pair of cells if the link rules inside these cells match each other's syntactic and semantic links without violating syntactic link order constraints of the link grammar.

In Figure 7, we have shown grammar rules and marking in the charts of two nodes 'before' and '5 clock cycle'. The rules in this figure are labeled as L_i (where i is an index to the rule). Matched link rules have the same L_i and opposite polarity of syntactic and semantic links. For example, in Figure 7, the L_6 link rule in the first row of 'before' node chart matches the L_6 link rule of '5 clock cycles' chart. These matched L_6 link rules do not violate the syntactic link order constraints and can be connected.

As illustrated in Figure 7 (c), we record the connection between these links by circling them in the cell. These links can be part of the final linkage when all the mandatory links in their rows are connected. For example, the L_6 link in the first row of the 'before' chart can be part of a final linkage if we find a matching link for the L_5 link of the first row.

Table 1: Grammar size for specifications tested

Total Words	Total Chart	Total rows
417	261	821

We continue to match the remaining links between the two charts. The L_6 link rule in the second row of the 'before' chart can also be matched with the L_6 link rule of the '5 clock cycles' chart. This connection can result in a new linkage that will include L_6 and L_9 links. Since the L_6 of the '5 clock cycles' chart is already marked, we will create a new row by replicating the L_6 cell of the '5 clock cycle' chart as shown in Figure 7 (d). The replicated L_6 cell is then connected with the L_6 link rule of the second row of the 'before' chart. Replication of marked cells creates a new set of unique linkages

As illustrated in Figure 7, the cells of connected links are recognized by assigning a unique cell connection id at the bottom right corner of these cells. A cell connection id is derived from the node and cells ids of the marked links. A node id (marked in red) represents the node position in the sentence, and a cell id (marked in black) is shown in each chart cell. For example, in '5 clock cycles' chart, the cell connection id of the L_6 link in the first row is 5261 (node 5 cell 2 and node 6 cell 1) and is different from the L_6 link of the second row. A unique cell connection id acts as a pointer to the connected cell and assists in traversing the connected cells while selecting the BINGO rows.

Step2 Finding BINGO Row: After marking all possible cells in the chart, we scan the charts to find BINGO rows that pass through the connected cells and contain chart rows that have all their mandatory cells marked. Figure 8 illustrates the output after each step of the parsing for the specification 'system should reach dataReady before 5 clock cycles'. In Figure 8 (a) , the initial charts are shown with a subset of grammar rules of Figure 9.

Figure 8 (b) shows lines that span rows containing marked cells. The green line cannot be a BINGO row since it covers an unmarked mandatory cell L_9+. The Blue dashed line (Linkage 2) in the figure is a BINGO row with incomplete coverage since it does not cover all the nodes of the sentence. A complete coverage BINGO row is found by the Black dashed line that contains all the connected cells with mandatory cells marked and passes through the chart of all the nodes. The output of parsing is shown in figure 8 (c), where the final linkage is represented by the links covered by the BINGO row of the Black dashed line (Linkage 1).

Semantic creation: BINGO row provides linkages after taking into account the context of each node in the sentence. Nodes are connected with links in a transition-based dependency parsing framework like (Nivre, 2003). When a link is created between two nodes using either left-arc or right-arc transitions, then the semantic propagation rule associated with that link is also executed. The execution of semantic propagation rules for every transition arc between nodes results in the creation of a final SE at the root node of the parse tree. The resulting SVA translated from the root node SE is shown in Figure 8 (d).

6 Evaluation

The BINGO framework is written in JavaScript and is executed in Node.js platform. All experiments were run on a machine with 1.8 GHz Intel Core i7-8550u processor and 16GB RAM. We evaluated the

```
the = D+;;child
      L1

system = ( { D-;;head } &  Ssys+;;child ) or
             L1             L2
      ({ D-;;head } & CO-;occur_frame:occur_when_before_clock:slot:same;head  & Ssys+;occur_frame:occur_when_before_clock:slot:same;child )
         L1                          L9                                                    L12

should = ( Ssys-;;head & I+;None:None:SE_SE:occur_frame;head )
           L2                         L3

reach  = ( I-;None:None:SE_SE:occur_frame;child & O+;occur_frame:occur_what_event:node;head  & MVp+;occur_frame:occur_when_before_clock:slot:same;head ) or
                     L3                                  L4                                               L5
      ( I-;None:None:SE_SE:occur_frame;child  & O+;occur_frame:occur_what_event:node;head ) or
                     L3                                  L4
      ( I-;None:None:SE_SE:occur_frame;child & O+;occur_frame:occur_what_event:node;head  & MVp+;occur_frame:occur_when_after_clock:slot:same;head )
                     L3                                  L4                                               L10

5 clock cycles  = (  J-;occur_frame:occur_when_before_clock:node;child ) or
                                      L6
                  ( J-;occur_frame:occur_when_after_clock:node;child )
                                      L11

dataReady = ( O-;occur_frame:occur_what_event:node;child   &  { ANE+;;head } )
                              L4                                 L7
state = ( ANE-;;child  )
            L7

before = ( MVp-;occur_frame:occur_when_before_clock:slot:same;child & J+;occur_frame:occur_when_before_clock:node;head )  or
                              L5                                                      L6
      (  J+;occur_frame:occur_when_before_clock:node;head          & CO+; occur_frame:occur_when_before_clock:slot:same;child   )
                              L6                                                      L9

on = ( MVp-;occur_frame:occur_when_before_clock:slot:same;child & J+;occur_frame:occur_when_before_clock:node;head ) or
                              L5                                                  L6
   ( MVp-;occur_frame:occur_when_after_clock:slot:same;child  & J+;occur_frame:occur_when_after_clock:node;head  )
                              L10                                                 L11
```

Figure 9: A small set of grammar rules to parse specifications

Table 2: Specification types, count and average JavaScript processing time for Specs

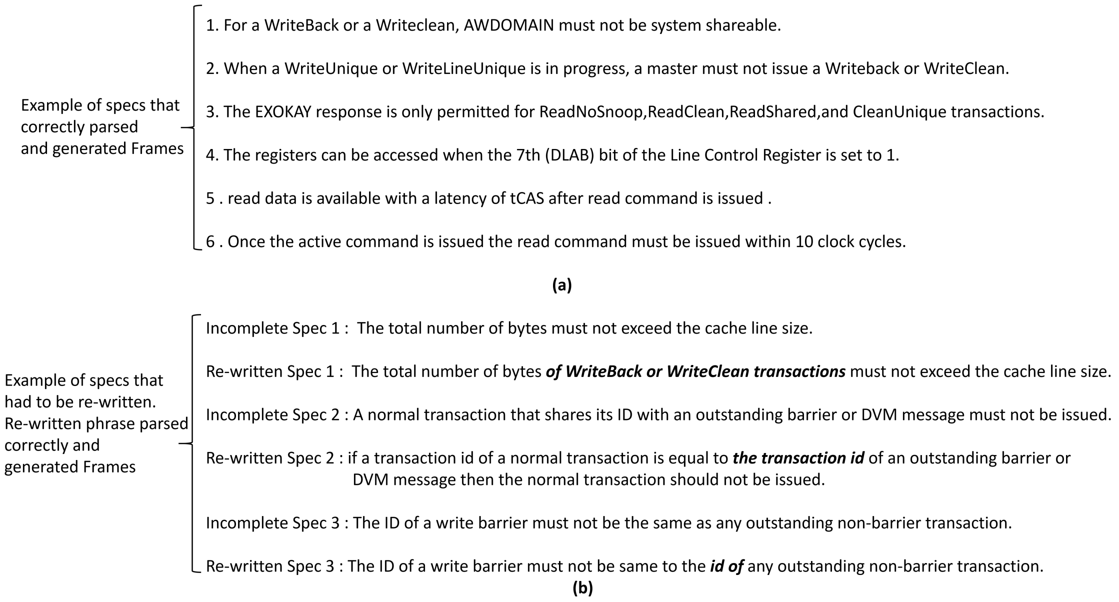
Spec type	Count of Spec	Avg. time
RTL Spec	123	245 ms
High Level Spec	113	526 ms
Memory controller Spec	40	101 ms
UART Spec	40	178 ms

framework by creating the grammar with syntactic and semantic link rules that can parse specifications found in documents. As shown in Table 1, our grammar consisted of a vocabulary of 417 words. We created 261 charts and 821 rows to represent syntactic and semantic connections of the words in our grammar. We present the framework's performance in Table 2 in terms of the type of specifications and number of specifications parsed, and the average time taken to parse each specification.

Similar to earlier approaches in (Harris and Harris, 2016; Zhao and Harris, 2019; Keszocze and Harris, 2019; Krishnamurthy and Hsiao, 2019), we picked specifications of ARM's AMBA protocol from (ARM, 2012) and (ARM, 2006) documents that have the names of all the signals and registers needed to generate an SVA code. In Table 2, RTL Spec type under the Spec type column refers to these low abstraction level specifications. We successfully created the grammar for 123 specifications of RTL Spec type and generated the corresponding SVA code. A small set of these specifications with the translated

SPECS	TRANSLATIONS
AWID must remain stable when AWVALID is asserted and AWREADY is low.	assert property (@(posedge clk) (AWVALID == 1) & (AWREADY == 0) |-> $stable(AWID));
A value of X on AWADDR is not permitted when AWVALID is high	assert property (@(posedge clk) (AWVALID == 1) |-> AWADDR != X);
When BVALID is asserted then it must remain asserted until BREADY is high.	assert property (@(posedge clk) (BVALID == 1) |-> $stable(BVALID)[*1:$] (BREADY == 1));
ACREADY should be asserted within MAXWAITS cycles of ACVALID being asserted	assert property (@posedge clk) (ACVALID == 1) |-> ##[1:MAXWAITS] ACREADY);

Figure 10: Specifications with RTL information are translated to SVA code.



Example of specs that correctly parsed and generated Frames

1. For a WriteBack or a Writeclean, AWDOMAIN must not be system shareable.

2. When a WriteUnique or WriteLineUnique is in progress, a master must not issue a Writeback or WriteClean.

3. The EXOKAY response is only permitted for ReadNoSnoop,ReadClean,ReadShared,and CleanUnique transactions.

4. The registers can be accessed when the 7th (DLAB) bit of the Line Control Register is set to 1.

5 . read data is available with a latency of tCAS after read command is issued .

6 . Once the active command is issued the read command must be issued within 10 clock cycles.

(a)

Example of specs that had to be re-written. Re-written phrase parsed correctly and generated Frames

Incomplete Spec 1 : The total number of bytes must not exceed the cache line size.

Re-written Spec 1 : The total number of bytes ***of WriteBack or WriteClean transactions*** must not exceed the cache line size.

Incomplete Spec 2 : A normal transaction that shares its ID with an outstanding barrier or DVM message must not be issued.

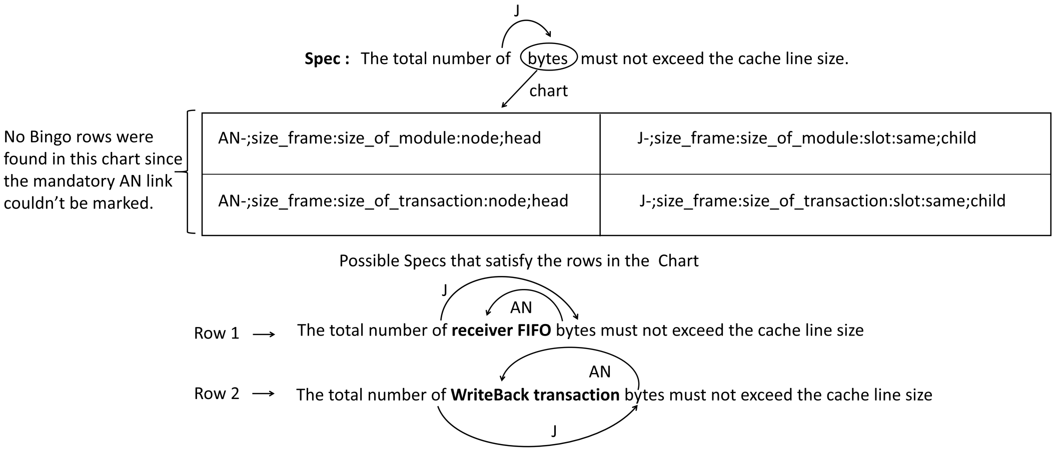
Re-written Spec 2 : if a transaction id of a normal transaction is equal to ***the transaction id*** of an outstanding barrier or DVM message then the normal transaction should not be issued.

Incomplete Spec 3 : The ID of a write barrier must not be the same as any outstanding non-barrier transaction.

Re-written Spec 3 : The ID of a write barrier must not be same to the ***id of*** any outstanding non-barrier transaction.

(b)

Figure 11: (a) Examples of specifications that were correctly parsed. (b) Example of specifications that had to be re-written for correct translations to semantic frames.



Spec : The total number of bytes must not exceed the cache line size.

AN-;size_frame:size_of_module:node;head	J-;size_frame:size_of_module:slot:same;child
AN-;size_frame:size_of_transaction:node;head	J-;size_frame:size_of_transaction:slot:same;child

No Bingo rows were found in this chart since the mandatory AN link couldn't be marked.

Possible Specs that satisfy the rows in the Chart

Row 1 ⟶ The total number of **receiver FIFO** bytes must not exceed the cache line size

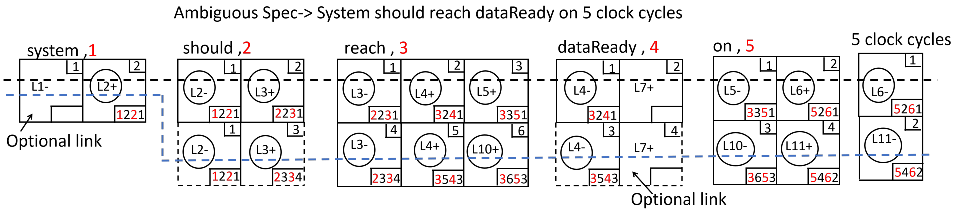
Row 2 ⟶ The total number of **WriteBack transaction** bytes must not exceed the cache line size

Figure 12: A BINGO row does not exist for incomplete specifications and we cannot create semantic frames for these specifications.

SVA is shown in Figure 10. However, as shown in (Krishnamurthy and Hsiao, 2020), these specifications are concise and lack variations in their sentence structures.

In order to evaluate the framework on different types of sentence structures, we created rules to generate semantic frames for high-level abstraction specs of the AMBA 4 ACE protocol checker document (ARM, 2012) . In the second row of Table 2, the High level Spec type represents specifications of a higher abstraction level that can only be translated to Frames due to the lack of low-level design variable names in these specs. We parsed a total of 113 high-level specs from AMBA 4 ACE (ARM, 2012) document. We further evaluated the tool by manually extracting and re-writing 40 memory controller specifications from (Vijayaraghavan and Ramanathan, 2006) and 40 specifications from UART (Gorban, 2002) documents. A small set of these specifications and the high-level specs of AMBA 4 ACE protocol that were translated to semantic frames are shown in Figure 11 (a).

In Table 2, the "Avg. time" column represents the average time taken to parse and create semantic frames for each spec. For example, it took 30.2 seconds to create semantic frames for all 123 RTL Spec, which gives an average time of 245 ms to parse each RTL spec. Specifications with many conjuncts took the maximum amount of time to parse. For example, the specification taken from (ARM, 2012):



(a) Two rows passes through all the marked cells

on = (MVp-;occur_frame:**occur_when_before_clock**:slot:same;child & J+;occur_frame:**occur_when_before_clock**:node;head) or
L5 L6
(MVp-;occur_frame:**occur_when_after_clock**:slot:same;child & J+;occur_frame:**occur_when_after_clock**:node;head)
L10 L11

(b) Grammar rules for the word "on"

Figure 13: Ambiguous spec generates more than one BINGO row.

"A slave must not give an Isshared (RESP[3] = 'b1) response to a readsnoop, readunique, cleanunique, cleaninvalid, makeinvalid or makeunique transaction" takes 2 seconds to parse.

Our framework could not infer data that was not explicitly present in the specification. These incomplete specifications were detected when no BINGO rows were generated. Figure 12 illustrates an example of an incomplete specification 'The total number of bytes must not exceed the cache line size.'. The specification does not explicitly specify a module name or a transaction whose bytes are being referred. As shown in this figure, the bytes could belong to a receiver FIFO module of UART or can be a part of the WriteBack transaction of the AMBA protocol. In Figure 12, the chart represents the syntactic-semantic connections for the node 'bytes'. In this chart, the node 'bytes' require an AN- syntactic link to fill either a size_of_module or size_of_transaction slot in the size_frame. In the example specification, the linkage requirements of AN- link for the node 'bytes' could not be satisfied, and no BINGO row was found for the spec. Consequently, no SE could be produced from this incomplete spec. We had to re-write the incomplete specifications with the complete information needed for accurate understanding. In Figure 11 (b), we have illustrated examples of incomplete specifications of (ARM, 2012) that we re-wrote with additional details. The re-written specs were accurately parsed and translated to semantic frames.

In our framework, ambiguous sentences can be detected when more than one BINGO row with complete coverage is generated in the parsing stage. Figure 13 shows an example of an ambiguous specification 'System should reach dataReady on 5 clock cycles.' The specification is ambiguous since it is unclear if the dataReady state should reach before or after 5 clock cycles. The node 'on' can connect with node '5 clock cycles' using two combinations of link rules as illustrated in the Figure 13 (a). The connection with link L_6 passes the node 5 clock cycles to the slot occur_when_before_clock, and the connection with link L_{11} propagates the 5 clock cycles to the slot occur_when_after_clock. The connections result in two different interpretations that cover all the nodes in the spec.

7 Conclusions and Future Work

In this paper, we demonstrated a dependency grammar-based framework to process hardware design specifications written in English. Our grammar is inspired from the syntactic link grammar. We have introduced contextual information and semantic propagation rules to the grammar using semantic links. We successfully evaluated the framework on specifications for a different range of hardware assertions taken from documents of four types of hardware architectures. We further modified some spec statements to test the robustness of the framework on handling different sentence structures. Our future work will further investigate the automatic detection of incomplete and ambiguous specifications and the generation of suggestions that can assist a user in writing specifications according to grammar.

Acknowledgement

The research is supported in part by National Science Foundation grant #2101021.

References

ARM, 2006. *AMBA 3 AXI Protocol Checker User Guide.* `https://developer.arm.com/documentation/dui0305/b/Assertion-Descriptions`.

ARM, 2012. *AMBA 4 ACE and ACE-Lite Protocol Checkers User Guide.* `https://developer.arm.com/docs/dui0576/b/ace-and-ace-lite-protocol-assertion-descriptions`.

Imran Sarwar Bajwa, Mark Lee, and Behzad Bordbar. 2012. Resolving syntactic ambiguities in natural language specification of constraints. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 178–187. Springer.

Aishwarya Chhabra, Amit Sangroya, and C Anantaram. 2018. Formalizing and verifying natural language system requirements using petri nets and context based reasoning. In *MRC@ IJCAI*, pages 64–71.

Michael A. Covington. 2001. A fundamental algorithm for dependency parsing. In *Proceedings of the 39th Annual ACM Southeast Conference*, pages 95–102.

Marie-Catherine De Marneffe and Joakim Nivre. 2019. Dependency grammar. *Annual Review of Linguistics*, 5:197–218.

Aaron Dutle, César Muñoz, Esther Conrad, Alwyn Goodloe, Laura Titolo, Ivan Perez, Swee Balachandran, Dimitra Giannakopoulou, Anastasia Mavridou, and Thomas Pressburger. 2020. From requirements to autonomous flight: An overview of the monitoring icarous project. In Matt Luckcuck and Marie Farrell, editors, Proceedings Second Workshop on *Formal Methods for Autonomous Systems,* Virtual, 7th of December 2020, volume 329 of *Electronic Proceedings in Theoretical Computer Science*, pages 23–30. Open Publishing Association.

Shalini Ghosh, Daniel Elenius, Wenchao Li, Patrick Lincoln, Natarajan Shankar, and Wilfried Steiner. 2016. Arsenal: automatic requirements specification extraction from natural language. In *NASA Formal Methods Symposium*, pages 41–46. Springer.

Dimitra Giannakopoulou, Thomas Pressburger, Anastasia Mavridou, and Johann Schumann. 2020. Generation of formal requirements from structured natural language. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 19–35. Springer.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

Jacob Gorban, 2002. *UART ip core specification Architectures.* `http://www.isy.liu.se/edu/kurs/TSEA44/OpenRISC/UART_spec.pdf`.

Xiaodong Gu, Hongyu Zhang, Dongmei Zhang, and Sunghun Kim. 2016. Deep api learning. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 631–642.

Christopher B Harris and Ian G Harris. 2016. Glast: Learning formal grammars to translate natural language specifications into hardware assertions. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 966–971. IEEE.

Michael S Hsiao. 2018. Automated program synthesis from object-oriented natural language for computer games. In *Proceedings of the Sixth International Workshop on Controlled Natural Language (CNL)*, pages 71–74.

Michael S Hsiao. 2021. Multi-phase context vectors for generating feedback for natural-language based programming. In *Proceedings of the Seventh International Workshop on Controlled Natural Language (CNL).*

Oliver Keszocze and Ian G Harris. 2019. Chatbot-based assertion generation from natural language specifications. In *2019 Forum for Specification and Design Languages (FDL)*, pages 1–6. IEEE.

Rahul Krishnamurthy and Michael S Hsiao. 2019. Controlled natural language framework for generating assertions from hardware specifications. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pages 367–370. IEEE.

Rahul Krishnamurthy and Michael S Hsiao. 2020. Transforming natural language specifications to logical forms for hardware verification. In *2020 IEEE 38th International Conference on Computer Design (ICCD)*, pages 393–396. IEEE.

Xi Victoria Lin, Chenglong Wang, Luke Zettlemoyer, and Michael D Ernst. 2018. Nl2bash: A corpus and semantic parser for natural language interface to the linux operating system. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).*

Anastasia Mavridou, Hamza Bourbouh, Dimitra Giannakopoulou, Thomas Pressburger, Mohammad Hejase, Pierre-Loic Garoche, and Johann Schumann. 2020. The ten lockheed martin cyber-physical challenges: formalized, analyzed, and explained. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 300–310. IEEE.

Zifan Nan, Hui Guan, Xipeng Shen, and Chunhua Liao. 2021. Deep nlp-based co-evolvement for synthesizing code analysis from natural language. In *Proceedings of the 30th ACM SIGPLAN International Conference on Compiler Construction*, pages 141–152.

Alexis Nasr and Owen Rambow. 2004. A simple string-rewriting formalism for dependency grammar. In *Proceedings of the Workshop on Recent Advances in Dependency Grammar*, pages 17–24.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the eighth international conference on parsing technologies*, pages 149–160.

Sandip Ray, Ian G Harris, Goerschwin Fey, and Mathias Soeken. 2016. Multilevel design understanding: from specification to logic. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–6. IEEE.

Daniel Sleator and Davy Temperley. 1991. Parsing english with a link grammar. carnegie mellon university computer science technical report cmu-cs-91-196. `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/link/pub/www/papers/ps/tr91-196.pdf`.

Daniel D Sleator and David Temperley. 1993. Parsing english with a link grammar. In *Proceedings of the Third International Workshop on Parsing Technologies*, pages 277–292.

Mathias Soeken, Christopher B Harris, Nabila Abdessaied, Ian G Harris, and Rolf Drechsler. 2014. Automating the translation of assertions using natural language processing techniques. In *Proceedings of the 2014 Forum on Specification and Design Languages (FDL)*, volume 978, pages 1–8. IEEE.

Srikanth Vijayaraghavan and Meyyappan Ramanathan. 2006. Sva for memories. In *A practical guide for SystemVerilog assertions*, chapter 5, pages 191–232. Springer US, Boston, MA. `https://doi.org/10.1007/0-387-26173-7_6`.

Rongjie Yan, Chih-Hong Cheng, and Yesheng Chai. 2015. Formal consistency checking over specifications in natural languages. In *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1677–1682. IEEE.

Shiyu Zhang, Juan Zhai, Lei Bu, Mingsong Chen, Linzhang Wang, and Xuandong Li. 2020. Automated generation of ltl specifications for smart home iot using natural language. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 622–625. IEEE.

Junchen Zhao and Ian G Harris. 2019. Automatic assertion generation from natural language specifications using subtree analysis. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 598–601. IEEE.

Drawing the syntactic space: choices in diagrammatic reasoning

Nicolas Mazziotta
Université de Liège
Département de Langues et littératures françaises et romanes
U.R. Traverses
nicolas.mazziotta@uliege.be

Abstract

This paper focuses on syntactic diagrams, i.e. formalized graphical representations (*inscriptions*) of dependency-based syntactic knowledge. In a simplified way, syntax can be described as the combination of three kinds of information: word order, dependencies and relations between equivalent terms. In the graphical space, pieces of information related to those different aspects combine. Hence, diagrams can be used as tools to investigate the interactions between them. Because of the graphical nature of the diagrams, some of their components are more salient than others. That fact implies that diagrams that represent similar information may differ in the contents to which they actually draw attention.

1 Introduction

This paper focuses on syntactic diagrams as tools to investigate linguistic materials. Although diagrams are part of the research process for many syntacticians, studies on their uses and on their semiotic properties are very sparse. With the development of treebanks, many tools have been implemented to provide visual representations of syntactic trees, but their elaboration is hardly ever discussed outside of private project meetings. In my opinion, representational conventions deeply interfere with research procedures. In several previous studies (some of them in collaboration with Sylvain Kahane), I focused on the formal, semiotic and grammatical aspects of diagrams (Kahane and Mazziotta, 2015; Mazziotta, 2019; Mazziotta, 2020). In this paper, I will mainly focus on the rhetorical consequences of diagrammatic choices.

The following study deals with the articulation between three different types of information in syntactic diagrams at use in dependency-based descriptions,[1] namely: word order, dependencies and relations between equivalent terms (coordinations and "paradigmatic piles"). I will adress representational choices that make it possible to visualize simultaneously different aspects of the analysis. More importantly, I will question the value of such simultaneous representations. In order to do so, Section 2 introduces the semiotic notion of *diagram*. Section 3 illustrates frequently used diagrams that express the three aforementionned types of information. Section 4 focuses on the use of diagrams in syntactic reasoning, and highlights the concepts of *salience* and *exhibitive efficiency*. Section 5 concludes by highlighting major points.

2 Diagrams, reification and configuration

According to C.S. Peirce, diagrams are formalized icons of representations (Stjernfelt, 2007, 90-102). They act as complex signs, and their internal structure is similar to the one of the contents they mean to represent. In this paper, the use of the term *diagram* will be limited to the specific meaning of "formalized graphical figures". I will follow the "theory of support" (Bachimont, 2007; Bachimont, 2010), according to which diagrams are devices (Fr. *dispositifs*) that are used to express and access knowledge. Diagrams are *inscriptions* of knowledge. The main point of this theory is that knowledge cannot be expressed

[1] ICA-based diagrams are not discussed in this paper.

unless it is inscribed, and that such inscriptions are diverse. In the case of syntactic analyses, they can be inscribed in the form of tree-like diagrams or in an algebraic form. Moreover, several concurrent diagrammatic representations can effectively express the same knowledge about syntactic structures.

Since diagrams are formalized graphical structures, they are constrained by formal rules. Since diagrams are inscriptions on a medium, they are constrained by the physical properties of the latter. Formal rules consist of the inventory of the graphical entities (discrete symbols) that can be used in the diagrams, and rules governing their organization on the plane, i.e. configurational rules that constraint the spatialization of the symbols. This can be illustrated with a dependency tree of (1) such as Fig. 1: 1/ orthographic word-forms represent words; 2/ strokes placed between the words represent dependencies (such strokes are frequently supplemented with labels, that I abtract away from the discussion for the sake of simplicity).

(1) The boy ate a cookie (Groß, 2003)

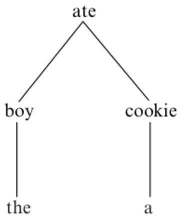


Figure 1: Diagrammatic inscription of a dependency tree (Groß, 2003, 331)

In my terms, syntactic concepts such as *words* and *dependencies* are *reified* by *graphical entities* (or *entities*, for short) (Groupe μ, 1992), i.e. discrete shapes. Syntactic concepts are not only inscribed by reification: reification is complementary to *configurational* rules regarding the relative spatialization of the graphical entities. Such configuration represents relations between terms of the syntactic analysis. In Fig. 1, *ate* appears higher on the plane than *boy*. This fact, combined with the fact that both terms appear at each ends of a stroke, express the fact that *ate* governs *boy*.

The limitations associated with the medium are very tangible. From a geometrical perspective, on the physical plane of a sheet of paper or on the screen of a computer, both the vertical dimension and the horizontal one must be present. Moreover, no additional geometric dimensions can be added to them. This limitation is crucial for the issues at study. Its main effect is that the contents that are inscribed in a diagram must be reduced accordingly.

3 Syntactic space and frequent diagrams

The complete *syntactic space* (henceforth *S*) corresponds to the set containing all the syntactic knowledge that can potentially be elaborated. I posit that *S* contains several subspaces (subsets), without specifying their exact number. This paper acknowledges a simplified syntactic space and focuses in particular on three of its subspaces, which correspond to aspects that frequently appear in diagrams: syntactic dependencies, word order and grouping of equivalent terms. For convenience, the three subspaces are considered axiomatic from the perspective of this paper.

Formally, it is possible to elaborate an algebraic structure that contains the whole syntactic information in a modular way – the objective of Mel'čuk's "functional" approach to synthesis is to elaborate such a structure (Mel'čuk, 2021, 9). Provided that the linguistic analysis is done, it would be trivial to encode an algebraic structure into a computational object in a programming language. Similarly, diagrams are often regarded as graphical representations of a preliminary algebraic formalisation. In cases such as automatic generation of diagrams for the purpose of extracting data from a treebank, diagrams are indeed second-rank signs, *transpositions* (Hébert, 2020, 143-144) of another inscription (an algebraic encoding). However, in many cases of theoretical syntactic reasoning, diagrams are primary formal inscriptions of the analysis in a specific medium: the graphical space (henceforth *G*). For theoretical linguists who use diagrams, synctactic theories are expressed through them, and would remain fuzzy and ill-defined without any kind of representation. Consequently, following the grounding assumptions of the theory of support, I consider that *S* is an abstract amorphous and unreachable knowledge that can be accessed only through an inscription. Conversely, the inscription of this knowledge in *G* is concrete, formalized and can be accessed.

Syntactic diagrams inscribe one or several subspaces of *S* in the graphical space *G*. The inscription of the subspaces grounds their formal definitions, that must conform to the limitations (notably, bidimensionality) of *G*, in which the inscriptions are embedded.

In this section, I briefly define the subspaces of dependencies (Subsection 3.1), word order (Subsection 3.2) and equivalences, i.e. grouping of equivalent terms (Subsection 3.3), and I illustrate the diagrams specifically used to inscribe them. It will quickly become obvious that diagrams often merge elements pertaining to different subspaces into a single inscription.

3.1 Dependencies

The subspace of dependencies (henceforth *D*) corresponds to the internal structure of so-called *rectional units*. Much could be discussed about the concept of *dependency*, its possible subdivision into "deep" and "surface" modules and the actual rules for identifying and classifying dependencies. This paper follows a classic general definition that remains implicit. *D* can be inscribed by the means of the rooted acyclic tree formalism. Each wordform is a node that appears exactly once as the second element of a couple of the tree, except for the single *root*, that only appears as the first element. Graphically, dependencies are inscribed in diagrams such as Fig. 1, which reads "*ate* governs *boy*", "*boy* governs *the*", etc. It is obvious that dependency trees like this one reify the relation of government by the means of strokes. In the case of Fig. 1, the vertical dimension is iconic of the hierarchy of the dependencies: governers are depicted higher than their dependents.

The tree formalism can encode branching relations. From the perspective of the inscription in *G*, branching necessitates the use of an additional dimension in order to avoid clashes between wordforms. Their reification must be spacialized in different positions on the plane.

3.2 Word order

Word order (henceforth *O*) corresponds to the encoding of the sequential order of the words in a well-formed oral or written construction (Tesnière, 2015, Chapters 5-9). For (1), *O* can be inscribed in the form of a *chain* (Mel'čuk and Milićević, 2014, 296-297) of words, as illustrated in Fig. 3.

the → boy → ate → a → cookie

Figure 2: Inscription of *O* in the form of a chain

In Fig. 3, arrows reify the relations of precedence between words. In Meaning-Text Theory (Mel'čuk, 1988, 48-49, 71), word order is encoded in the morphological module ("deep-morphological structure"), and precedence relations are not reified:

> [The] arcs [of the morphological structure] are, so to speak, degenerated; they specify only the strict linear ordering of wordforms ("$\mathbf{w_1}$ immediately precedes $\mathbf{w_2}$"), so that they need not be indicated explicitly. (Mel'čuk, 2009, 7)

In most cases, *O* is simply inscribed by the linear arrangement of orthographical symbols on the horizontal axis, that is, by a configurational convention, on a single dimension of *G*. The *O* subspace corresponds to the traditional inscription of written wordforms.

3.3 Equivalent terms and constructions

The third subspace, which I suggest to identify as the one of *equivalences* (henceforth *E*) contains information about equivalent terms and constructions. Contrary to *O* and to *D*, this subspace is not related to all the words of the analyzed sentence. *E* is less universally acknowledged, and requires a more detailed introduction.

The adjective *equivalent* expresses that some words can be grouped with respect to the fact that they can be substituted to each other in the same syntactic position. That is the case when words are involved in coordination, such as *hooded* and *armed* in (2). In diagrams, coordination is often inscribed alongside dependencies.

(2) hooded and armed youngsters (Kahane et al., 2019, 74)

Section 4 will focus on simultaneous inscriptions. In this subsection, I will introduce a type of diagram specifically used to inscribe equivalence between terms. In several projects traditionnaly related to the description of spoken French, such as *Rhapsodie* (Lacheret et al., 2019), phenomena such as repetitions and dysfluencies are also described as a grouping of equivalent terms (Kahane et al., 2019). Since equivalent terms share syntactic properties, and can be classified as members of the same paradigm, such groupings are sometimes called *paradigmatic piles*.

During the 90s, French scholars who focused on the description of spoken French came up with the idea of grid-like diagrams that make use of the vertical dimension of *G* to generate an iconic visualisation of the paradigmatic piles (Blanche-Benveniste and Jeanjean, 1986), including coordinations (Bilger, 1999). An example of such a diagram is provided in Fig. 4.

	hooded	
and	armed	youngsters

Figure 3: Inscription of *E* in the form of a grid

O is represented iconically if no equivalence has to be inscribed, but *hooded* and *armed* are spacialized one above the other in the vertical dimension.

As illustrated in (3), *Rhapsodie* (Lacheret et al., 2019; Kahane et al., 2019) adopts an alternative to the grid-like inscription: *E* is expressed by special entities (‘{’, ‘}’, ‘|’ and ‘ˆ’) that correlate the elements of paradigmatic piles and their boundaries with the order of the words (*O*).

(3) { hooded | ˆand armed } youngsters (Kahane et al., 2019, 74)

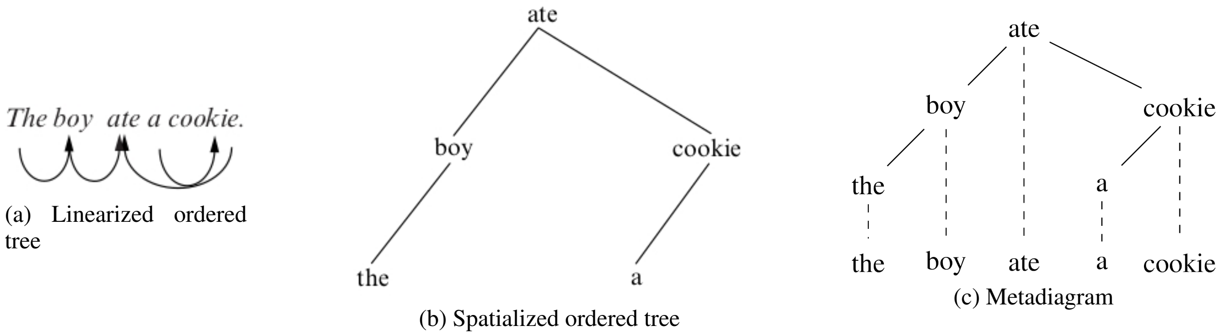
Those entities reify the limits of each pile in *G*. Configurationally, they interact with the rules related to the traditional inscription of *O*, because they appear on the same horizontal line. What makes it possible for *O* and *E* to be inscribed in the same dimension is the use of the space between words to reify elements that are not words. The reader must learn to interpret that curly braces function in pairs. Consequently, they allow for the inscription of recursive structures.

4 Joint inscriptions

The complete syntactic space *S* contains at least $D \cup O \cup E$ (notation: *DOE*). As illustrated in Subsection 3.3, traditional inscriptions of these subspaces in *G* often mix information from several of them, even if the focus is clearly on a single one. Diagrams are often grounded in polysemiotic systems (Hébert, 2020, 335 sv.): they express different contents according to different semiotic rules of interpretation. Consequently, diagrammatic inscriptions have to be built and interpreted by taking into account the relative *salience* of their components. Subsection 4.1 focuses on the joint inscription *DO* to illustrate that. On the other hand, joining the inscriptions of different subspaces on *G* allows for visual reasoning and visual investigations that take advantage of the polysemiotic environment (Subsection 4.2). In Subsection 4.3, I focus on the inscriptions of *DO* and *DOE* to show that not all representations are equivalent with respect to their exhibitive efficiency.

4.1 Polysemiotics, efficiency and salience

Since the most early uses of diagrams that inscribe at least some part of the dependency structure (Mazziotta, 2020; Osborne, 2020), *O* has often been clearly separated from *D*. Tesnière’s *Elements of structural* *syntax* are often cited as an important milestone in this respect (2015, Chapters 6 and 7) and major approaches such as Meaning-Text Theory abstract word order away from their syntactic modules (Mel'čuk, 2021, Chapter 10, e.g.). However, for the sake of ergonomics or for theoretical reasons, some dependency linguists consistently inscribe *DO* in *G*. Several frequent ways to do it are illustrated in Fig. 4 and in (4).



(a) Linearized ordered tree

(b) Spatialized ordered tree

(c) Metadiagram

Figure 4: Alternative inscriptions of *DO*: (a) (Groß, 2003, 332); (b) (Groß, 2003, 332); (c) (Groß, 2003, 334), modified according to Osborne's conventions (Osborne, 2019)

Fig. 4a illustrates a type of diagram that is extremely frequent for visualizing treebanks. However, for several scholars (Groß, 2003, 331) or (Osborne, 2019, 63), it lacks clarity. Indeed, the choice of one type of diagram or another depends on its *efficiency* with respect to the tasks/reading habits of the users (Bertin, 2005, 139), i.e. the relative speed at which they will be able to extract the piece of information they focus on. In Fig. 4a, the inscription of the hierarchy of syntactic dependencies (*D*) is made by the means of a symbolic reification: the arrowheads. The diagram makes a very restricted use of the vertical dimension, only to preserve the discreteness and readability of the arrows that reify them: the tree is linearized on one dimension. Therefore, the iconic inscription of the hierarchy that appears in Fig. 1 as well as in Fig. 4b and Fig. 4c is absent (hence the lack of clarity in Gross's and Osborne's opinion). The perception of the global structure of the dependencies is not the focal point of the diagram. From a phenomenological perspective, the inscription of *O* is more *salient* than the one of *D* (Hébert, 2020, 355-358).

Fig. 4b displays the oposite choice. Although the diagram also inscribes *DO*, the reader will find it difficult to extract *O*, because the graphical entities that reify the words are not aligned on a horizontal line. Instead, their position on the vertical axis depends on their status in *D*. The inscription of *D* is more sailent since the tree is spacialized. If the intent of the reader is to access information related to *O*, the vertical dimension is not interesting. From a semiotic perspective, it consists only of *noise* (Hébert, 2020, 45).

Fig. 4c can be considered as a composition of one diagram containing only information from *O* and a diagram of the kind of Fig. 4b, where *D* is more salient. The saliences of both diagrams are similar, but the distance between them on *G* makes it possible for the reader to focus on either one aspect or the other. Fig. 4c can be classified as a *metadiagram*: on the one hand, both diagrams are represented conjointly, and, on the other hand, by reifying the projection of the words implied in *D* on the linear axis expressing *O* (Groß, 2003, 334), the dashed strokes perform as inscriptions of relations between diagrams. In this case, the inscription of *DO* in *G* overtly states the relations between the two subspaces of *S*.

While it is not very common, *DO* can also be inscribed using bracketing conventions (Osborne, 2019, 61-63), as illustrated in (4).

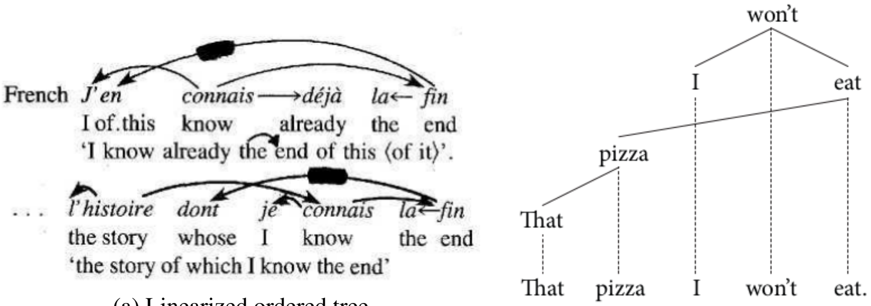
(4) [[[the] boy] ate [[a] cookie]]

Bracketing introduces special entities ('[' and ']') that inscribe *D* alongside *O* in the same horizontal dimension by means of configurational rules. The entities are interpreted in pairs as reifications of the limits of the dependency tree and its subtrees – similarly to the entities used in (3). Within each span delimited by a pair of entities, the governing word is not surrounded by brackets. In such cases, word order is very salient, and the embedding of many subtrees can lead to difficulties in identifying each pair of entities. Moreover, since the brackets must appear within the same dimension as the chain of words that inscribes *O*, the diagram cannot express projectivity violations without the introduction of additional conventions, which are illustrated in the next subsection.

4.2 Heuristics and joint inscriptions

According to C.S. Peirce, diagrams make creative reasoning possible (Stjernfelt, 2007, 102-107). One way to use diagrams is to manipulate them in order to discover new properties of the concepts inscribed in them. From a practical perspective, one can either keep subspaces of *S* apart from each other, or decide to merge them in *G*. In this respect, decisions taken are constrained by the desired ergonomics and by representational habits, but they also depend on the objectives of the research program; e.g. describing the interferences between the subspaces.

The limitations of *G* can be exploited heuristically. The joint inscription of *DO* commented in Subsection 4.1 is a convenient way to visualize projectivity violation (Ihm and Lecerf, 1963, 10) (Fig. 5).


French J'en connais ⟶ déjà la ← fin
I of.this know already the end
'I know already the end of this (of it)'.
. . . l'histoire dont je connais la fin
the story whose I know the end
'the story of which I know the end'
won't
I
eat
pizza
That
That pizza I won't eat.

(a) Linearized ordered tree

(b) Metadiagram

Figure 5: Heuristic use of inscriptions of *DO*: (a) (Mel'čuk, 1988, 37); (b) (Osborne, 2019, 204)

Fig. 5a makes use of conventions that are similar to the ones of Fig. 4a: both graphical dimensions are used to encode *D*: horizontally to encode the extremities of the dependencies and vertically to preserve the distinctiveness and the discreteness of the arrows. Since *O* is not inscribed in a separate graphical dimension, there is no way to prevent arrows from crossing each other in cases of projectivity violation. Non-projective structures can only be inscribed by crossing dependency arrows. In Fig. 5b (that follows the same conventions as Fig. 4c), projectivity violation are made even more explicit, since they correspond to the fact that dependency strokes cross the dashed lines that reify the correspondances between *D* and *O*. That is, due to the geometric properties of the plane, dependencies cross projection lines. The interaction is inscribed in a completely iconic way. Separate inscriptions do not contain that part of information on the data that emerges geometrically from their joint inscription.

4.3 Exhibitive efficiency

Diagrams allow graphical reasoning because the graphical inscription of syntactic contents *exhibits* structures and their interactions. The term *exhibit* corresponds to the fact that diagrams have purposely salient elements that are meant to be focalized by the reader, i.e. they have a rhetorical orientation. It is noteworthy to highlight that inscriptions such as Fig. 5 convey superfluous information for the reader who has no interest in projectivity: for them, lines crossing each other are undesirable noise. In such cases, diagrams exhibit information that the reader does not want to examine. As we have seen, the salience of the graphical entities and their configurations is more or less efficient in order to discover the linguistic properties of units pertaining to several subspaces of *S*.

Tesnière insists that *O* must be abstracted away from *D* and *E*. He suggests that coordinations and appositions are somewhat "orthogonal" to dependencies (Kahane, 2012). This orthogonality corresponds to a different dimension: Tesnière inscribes the part of *E* that corresponds to coordination directly in a dependency structure that does not encode linear order.[2] In his view, coordination duplicates dependencies without affecting the valency of the governor (Tesnière, 2015, Chapter 135). Fig. 6 expresses this interaction between *D* and *E*.

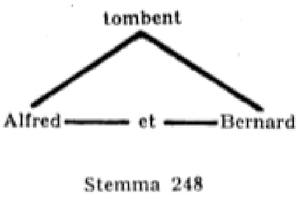


Figure 6: Inscription of *DE* (Tesnière, 1966)

The expression of *O* holds no theoretical basis and is only incidentally expressed (Mazziotta, 2019, 73). Hence, the diagram can use the horizontal dimension to inscribe two kinds of information: 1/ the distinction between elements from *D* (diagonal strokes are discrete and distinct from each other); 2/ the paradigmatic relation of *E* is reified by a horizontal stroke labeled with a conjunction.[3] Tesnière's conventions exhibit the orthogonal nature of the relations between the two subspaces.

Following Tesnière, several attempts to encode *DE* and even *DOE* have been made. I will now focus on a sample selection and investigate their exhibitive efficiency. For instance, Fig. 7 uses pseudo-tridimensional conventions to show that *D* and *E* belong to different planes.

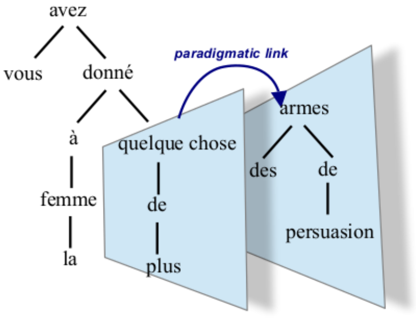


Figure 7: Inscription of *DE* (Kahane et al., 2019), original diagram provided by S. Kahane

Despite Fig. 7 being an hapax (such inscriptions are not generalized), it is crucial to my point: in this specific case, *DE* are inscribed, but the representational choices actually make *E* (front) more salient than *D* (back). The diagram exhibits the structure of equivalences, under the name of *paradigmatic piles*, precisely in a chapter that focuses on explaining the identification principles and the annotation procedures of paradigmatic piles. From a rhetorical perspective, the diagram is efficient.

Let us examine another case. Tesnière's diagrams are unable to assess recursion because the stroke that reifies the paradigmatic pile is spacialized on a single dimension. Osborne proposes diagrammatic conventions that solve this problem (2019, Chapters 10 and 11). He considers that the structure of *E* is actually constituency-based. Therefore, it justifies that the entities and configurations used to inscribe this subspace are different from the ones used to inscribe *D*: equivalences are reified by angled stokes and, with some redundancy (Hébert, 2020, 346-347), by squared brackets that interact with *O* as curly braces do in Fig.3.

The diagram simultaneously inscribes *DOE*: *O* and *D* are expressed as explained in Subsection 4.2; the entities used to express *E* obey configurational rules that make it possible to encode recursion. That kind of diagram, although it conveys a great amount of information, leaves to the reader the responsibility to

[2] Similar solutions had already been introduced by several American (Mazziotta, 2020) and German (Osborne, 2020) scholars in the 19th century.

[3] Tesnière explains that the conjunction is not connected to the conjuncts by two separate strokes: it labels a single stroke by interrupting it (Mazziotta, 2014).

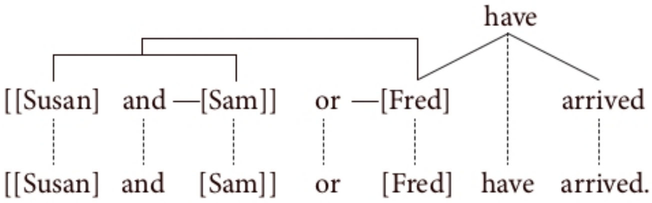

have
[[Susan] and —[Sam]] or —[Fred] arrived
[[Susan] and [Sam]] or [Fred] have arrived.

Figure 8: Inscription of *DOE* (Osborne, 2019, 323)

focus on various elements of similar salience. For instance, it expresses that the word *Fred* is part of *E* as well as a part of *D*, but it *does not exhibit it*. The reader has to navigate the diagram to evaluate it. They can understand by themselves that the perspective is not the same as Tesnière's. Osborne considers that the members of equivalent relations are parallelized (2019, 324, in partic. note 247). Some conjuncts do not have an ancestor in the diagram: in such cases, the governor needs to be reconstructed by comparing other conjuncts until a governor is found. In Tesnière's diagram, dependency strokes are multiplied (2015, Chapter 135), whereas in Osborne's, they are not. None of this is purposely exhibited in the diagram.

Fig. 9 also inscribes simultaneously *DOE*, but the difference between *D* and *E* is symbolic. Arrows are classified by the means of contrasts between colors and oppositions between plain and dotted strokes. Although all three subspaces are inscribed, if the objective of the reader is to process interactions between them, the diagram is even more cumbersome to handle than Fig. 4a is. However, it exibits that subspaces can be efficiently encoded in a unique formalism from the perspective of the elaboration of treebanks.

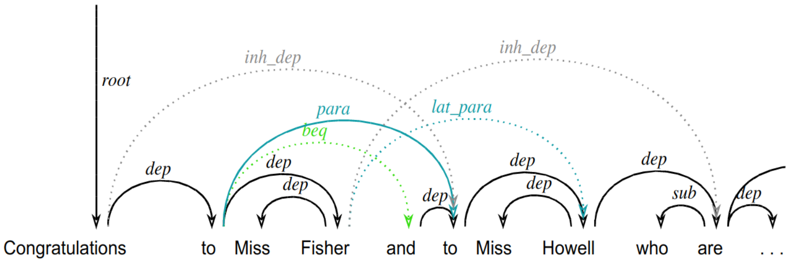


Figure 9: Inscription of *DOE* (Gerdes and Kahane, 2015)

As illustrated, there exist different ways to inscribe simultaneously the syntactic subspaces. However, even if information is similar in different diagrams, they do not exhibit the same contents. Since diagrams range from the most salience-neutral choices to the most rhetorically oriented, choosing between them in order to select the right one is crucial.

5 Conclusion

I have described the syntactic space (*S*) as a combination of three subspaces: dependencies (*D*), word order (*O*) and equivalences (*E*). In Section 3, I have introduced these subspaces and the diagrams frequently used to inscribe them in the graphical space (*G*). I have pointed out that, unless it is inscribed, *S* remains abstract and amorphous: one can model information only by communicating knowledge through an inscription. Graphical inscriptions of syntactic concepts combine graphical entities that reify conceptual units with configuration rules that govern the spatialization of those entities. They are genuine formalisms, relying on an inventory of units and rules that govern them.

On the other hand, the materiality of *G* greatly impacts how diagrams are elaborated and their practical exploitation. In Subsection 3.3, it became obvious that several subspaces of *S* are inscribed simultaneously in the same diagram. In Section 4, I have explored several cases of joint inscription of two or three subspaces in order to describe their heuristic power. Such diagrams pertain to polysemiotics that involve parts that can be contrasted with respect to their relative salience (Subsection 4.1). The polysemiotic nature of the diagrams make it possible to discover new pieces of knowledge on the interactions between the subspaces of S (Subsection 4.2). In the last part (Subsection 4.3), I have insisted that diagrams are actually rhetorically directed by their exhibitive intent. Consequently, they must be elaborated (and interpreted) by taking into account their practical use. Otherwise, they remain "noisy" inscriptions that transmit no relevant information.

Acknowledgements

I would like to thank Sylvain Kahane, András Imrényi, Julie Glikman, as well as the three anonymous reviewers of the first version of this paper for their comments and suggestions.

References

Bruno Bachimont. 2007. *Ingénierie des connaissances et des contenus. Le numérique entre ontologies et documents*. Hermès, Paris.

Bruno Bachimont. 2010. *Le sens de la technique: le numérique et le calcul*. Les belles lettres, Paris.

Jacques Bertin. 2005. *Sémiologie graphique. Les diagrammes – les réseaux – les cartes*. EHESS, Paris, 1st edition 1967; 4th edition.

Mireille Bilger. 1999. Coordination : analyses syntaxiques et annotations. *Recherches sur le français parlé*, 15:255–272.

Claire Blanche-Benveniste and Colette Jeanjean. 1986. *Le français parlé. Transcription et édition*. Didier, Paris.

Kim Gerdes and Sylvain Kahane. 2015. Non-constituent coordination and other coordinative constructions as dependency graphs. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 101–110, Uppsala, Sweden, August. Uppsala University, Uppsala, Sweden.

Groupe μ. 1992. *Traité du signe visuel*. Seuil, Paris.

Thomas Groß. 2003. Dependency grammar's limits – and ways of extending them. In Vilmos Ágel, Ludwig M. Eichinger, Hans-Werner Eroms, Peter Hellwig, Hans Jürgen Heringer, and Henning Lobin, editors, *Dependency and valency: An international handbook of contemporary research, Vol. 1*, pages 331–351. de Gruyter, Berlin and New York.

Louis Hébert. 2020. *Cours de sémiotique. Pour une sémiotique applicable*. Garnier, Paris.

P. Ihm and Yves Lecerf. 1963. Éléments pour une grammaire générale des langues projectives. Technical report, EURATOM, Ispra.

András Imrényi and Nicolas Mazziotta, editors. 2020. *Chapters of dependency grammar. A historical survey from Antiquity to Tesnière*. John Benjamins, Amsterdam and Philadelphia.

Sylvain Kahane. 2012. De l'analyse en grille à la modélisation des entassements. In Sandrine Caddeo, Marie-Noëlle Roubaud, Magali Rouquier, and Frédéric Sabio, editors, *Penser les langues avec Claire Blanche-Benveniste*, pages 101–116. Presses de l'Université de Provence, Aix-en-Provence.

Sylvain Kahane and Nicolas Mazziotta. 2015. Syntactic polygraphs. a formalism extending both constituency and dependency. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 152–164, Chicago, USA, July. Association for Computational Linguistics.

Sylvain Kahane, Paola Pietrandrea, and Kim Gerdes. 2019. The annotation of list structures. In Lacheret et al. (Lacheret et al., 2019), pages 69–95.

Anne Lacheret, Sylvain Kahane, and Paola Pietrandrea, editors. 2019. *Rhapsodie. A prosodic and syntactic treebank for spoken French*. John Benjamins, Amsterdam and Philadelphia.

Nicolas Mazziotta. 2014. Nature et structure des relations syntaxiques dans le modèle de Lucien Tesnière. *Modèles linguistiques*, 35:123–152.

Nicolas Mazziotta. 2019. The evolution of spatial rationales in Tesnière's stemmas. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, pages 69–80, Paris, France, August. Association for Computational Linguistics.

Nicolas Mazziotta. 2020. Dependency in early sentence diagrams: Stephen W. Clark. In Imrényi and Mazziotta (Imrényi and Mazziotta, 2020), pages 133–162.

Igor Mel'čuk and Jasmina Milićević. 2014. *Introduction à la linguistique. Volume 3*. Hermann, Paris.

Igor Mel'čuk. 1988. *Dependency syntax: theory and practice*. State University of New York, Albany.

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

Igor Mel'čuk. 2021. *Ten studies in dependency linguistics*. de Gruyter, Berlin and Boston.

Timothy Osborne. 2019. *A dependency grammar of English. An introduction and beyond*. John Benjamins, Amsterdam and Philadelphia.

Timothy Osborne. 2020. Franz kern: An early dependency grammarian. In Imrényi and Mazziotta (Imrényi and Mazziotta, 2020), pages 190–213.

Frederik Stjernfelt. 2007. *Diagrammatology. An investigation on the borderlines of phenomenology, ontology, and semiotics*. Springer, Dordrecht.

Lucien Tesnière. 1966. *Éléments de syntaxe structurale*. Klincksieck, Paris, 1st edition 1959; 2nd edition.

Lucien Tesnière. 2015. *Elements of structural syntax* [translation of (Tesnière, 1966) by T. Osborne and S. Kahane]. John Benjamins, Amsterdam and Philadelphia.

Causation (and Some Other) Paraphrasing Patterns in L1 English. A Case Study

Jasmina Milićević

Abstract

The present paper reports on a study aimed at testing the coverage of the Meaning-Text paraphrasing system by applying its rules to account for eighty paraphrases of an English sentence produced by eight native speakers of that language. We focus on "deep" paraphrasing links, in particular causation links, which can only be laid bare through semantic decompositions of lexical units involved. The results of the study corroborate the initial assumption that most of the paraphrasing links found in our mini corpus can be described in terms of the already existing paraphrasing rules. New paraphrasing rules are proposed for the small number of paraphrasing links hitherto unaccounted for.

1 Introduction

We start by characterizing the core linguistic concepts used in the study reported in the paper (Section *1.1*) and presenting the study goals (Section *1.2*).

1.1 Paraphrasing and its Modeling

As a particular case of synonymy, paraphrase, or (near-)synonymy of sentences, plays a crucial role in language acquisition and use (Žolkovskij & Mel'čuk 1967: 177, Fuchs 1980: 354ff, Matinot 2009, among others). An average speaker of language **L** is capable of producing and recognizing sentences of **L** that stand in the relation of paraphrase. Such sentences, illustrated in (1), are called (mutual) paraphrases, and the operation whereby they are produced is known as paraphrasing. (In this article, the converse operation of paraphrase recognition will be left aside.)

(1) a. *John's comfortable income enables him to travel frequently.*
b. *John's comfortable income allows him to travel often.*
c. *John makes a lot of money; therefore, he is able to travel a lot.*

Paraphrasing is recurred to both in everyday linguistic exchanges and in more complex writing and translation tasks—whenever there is a need to make oneself clearer, change one's style or find a more felicitous expression for the meaning to be conveyed. It is only normal, then, that paraphrases and their production are of great interest for theoretical linguistics, applied linguistics, translation and, more broadly, philosophy of language, logic and various other disciplines.

One of the most advanced models of paraphrasing proposed to date is the Meaning-Text paraphrasing system, introduced in Žolkovskij & Mel'čuk 1967 and further developed in several publications to be mentioned below: a set of rules that establish (quasi-)equivalences between linguistic elements (semantemes within meaning configurations, lexical units within syntactic constructions, and so on) at specific levels of representation of sentences recognized by Meaning-Text linguistic models. Meaning-Text paraphrasing system is characterized by an extensive coverage of linguistic phenomena involved in paraphrasing and cross-linguistic validity, thanks to the universally applicable formalisms used to formulate its rules.

There are two major types of paraphrasing rules. Lexical-syntactic paraphrasing rules, which operate at the deep-syntactic representation level, are based on lexical relations between lexical units involved—semantic-derivational (in a broad sense) and collocational—and syntactic constructions within which they appear. They treat quite sophisticated paraphrases, albeit less “deep” than those covered by semantic paraphrasing rules, which operate at the semantic level of representation and account for paraphrases that require semantic decomposition of lexical units present in the corresponding sentences. In this paper, we will see examples of both these rule types, while focusing on the latter, less well-developed and not as well-known as the former.

1.2 Goals of the Study

Our study intended to test the coverage of the Meaning-Text paraphrasing system by applying its rules to describe paraphrastic links in a corpus of paraphrases produced by native speakers of English.

The corpus consisted of eighty paraphrases of sentence (1a) above, organized around a causation verb (*to enable*) linking two facts (‘John has a lot of money’ and ‘John travels a lot’), chosen for its rich paraphrastic potential. The paraphrases were produced by the fourth year anglophone university students superficially initiated into the concept of paraphrasing and given minimal instructions as to how to go about the task at hand.

The immediate goals of the study were theoretical in nature, namely:

- look into paraphrastic diversity in the corpus, i.e., lexical and syntactic paraphrasing means used, and limits of paraphrastic variation;
- determine whether the existing Meaning-Text paraphrasing rules can account for the paraphrases in this particular corpus;
- if necessary, suggest new paraphrasing rules.

The assumption, corroborated by the study results, was that many, but not all, paraphrases in the corpus can be described in terms of the already existing paraphrasing rules.

No attempt was made to evaluate paraphrasing competence of the speakers (this task was left for a future study).

2 Global Analysis of the Corpus of Paraphrases

2.1 Quality of the Paraphrases

All sentences in the corpus were paraphrases of (1a) and grammatically correct; a few contained redundancies and/or stylistically marginal lexical choices, for instance:

(2) a. *The wages John experiences*$_{\text{[choice of the collocate]}}$ → *is paid give him a lot of travel opportunity.*
[Better: *The pay J. receives <earns> …*]
b. *John’s revenue*$_{\text{[paradigmatic lexical choice]}}$ → *income, earnings is/are such that he can travel often without economic*$_{\text{[paradigmatic lexical choice]}}$ → *financial pressure.*

Some paraphrases featured elements of pragmatic knowledge; i.e., they were not strictly linguistic paraphrases of (1a); here are some such pragmatic equivalences:

(3) a. [travel] *a lot* ~ [travel] *several times a year*
b. *travel* ~ *go on vacations* ; *take excursions* ; *get to see the world*
c. [allow …] *to travel* ~ [permit] *the luxury of travelling*
d. *enable* [to travel] ~ *grant the freedom* [to take trips]

In most cases, paraphrastic variation involved the propositional content and/or communicative orientation, a.k.a. information structure (Mel’čuk 2001; Féry & Ishihara 2016, eds); stylistic variation was present to a lesser degree. Both the global variation (affecting the overall organization of the sentence) and the local one (affecting individual sentence elements) were present.

The paraphrases in the corpus were lexically rich and structurally diverse. The vast majority were approximate paraphrases, with different degrees of semantic proximity: from very close to quite distant.

2.2 Paraphrastic Variation Found in the Corpus

Paraphrastic variation was measured according to two parameters, indicated in Table 1:

DIMENSIONS OF MEANING INVOLVED	SCOPE OF VARIATION
propositional content	global
communicative orientation	local
style (register)	

Table 1: Parameters used to determine paraphrastic variation in the corpus

Two paraphrases can differ along any or all three dimensions of meaning. With each dimension of meaning, the scope of variation can be either global or local. The variation can target one or more elements of the initial sentence. Consequently, several paraphrasing rules of different types may be necessary in order to produce just one pair of paraphrases.

On the one hand, paraphrastic variation is correlated to the depth of the paraphrasing link: generally speaking, the more radical the variation, the deeper the level of sentence representation at which it can take place. On the other hand, paraphrastic variation correlates with the exactness of the paraphrasing link: the more variation there is, the greater likelihood to get an approximate paraphrase of the starting sentence.

Global paraphrastic variation found in our corpus is represented in Tables 2 and 3 (for a full list of paraphrases, see Appendix):

	2-CLAUSE REALIZATION P = ‘John has money’; Q = ‘John can travel’ Main fact: the one implemented as the main predication/matrix clause. Completive clauses and nominal relative clauses are considered clause elements (parts of clauses).[1]	
A	Coordination	
	P: corroboration Q	*John must make a ton of money: he goes on vacation all the time.*
	P; consequently Q	*John makes a lot of money; therefore, he is able to travel frequently.*
B	Subordination	
B1	Main fact: P	
	P, which $Q_{\text{‘cause’}}$ <P, which causes Q>	*John makes good money, which permits him to take a lot of trips.* *John is paid well, which is why he can go on so many vacations.*
B2	Main fact: Q	
	Q caused by P	*John can travel lots since he has a comfortable income.* *John goes on a lot of trips because he’s well off.*
	Not Q if not P	*If John didn’t make as much as he does, he wouldn’t travel as often.*

Table 2: Two-clause realizations of paraphrases

	1-CLAUSE REALIZATION	
C1	Main fact: causation	
	P causes Q	*John’s earnings allow him to travel a lot.*
	Q is caused by <is a result of> P	*John’s frequent vacations have been enabled by his comfortable salary.* *Being able to travel a lot is a result of John’s good salary.*
C2	Main fact: P	
	P suffices for Q	*John earns enough money to afford to travel often.* *The money [that John makes] is sufficient to pay for all the trips he takes.*
	P is a reason for Q	*The reason why John is able to take so many trips a year is his great revenue.*
C3	Main fact: Q	
	Q is caused by <linked to> P	*John can travel lots since he makes a comfortable amount of money.* *Due to a nice income, John goes on many trips.* *John’s frequency of travel is correlated to his high remuneration.*

Table 3: One-clause realizations of paraphrases

2.3 Semantic Proximity of the Paraphrases in the Corpus

Example (4) illustrates different degrees of semantic proximity between the starting sentence (1a), repeated here as (4a), and some of its paraphrases from our corpus.

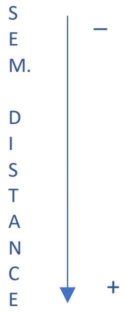
[1] An example of a nominal relative clause (bolded), which is a part of the syntactic subject of the sentence: *The money* [***that John makes***] *is sufficient to* …. In contrast, a sentential relative clause, for example, *John makes good money,* [***which permits him to …***] counts as a clause in its own right. For these terms, see Quirk *et al*. (1985: 1118*ff*).

(4) a. *John's comfortable income enables him to travel frequently.*
b. *His comfortable income allows John to travel often.*

c. *Thanks to John's substantial income, he is able to travel often.*
d. *With a comfortable income, John goes travelling quite a bit.*

e. *John is able to travel often because of his comfortable income.*
f. *John goes on a lot of trips because he's well off.*

g. *John must make a ton of money: he goes on vacation all the time.*
h. *How often John travels tells us that he makes a good living.*
i. *John's frequency of travel is correlated to his high remuneration.*



Sentences (4a-b) are exact mutual paraphrases; they do not differ with respect to any dimension of meaning (in other words, their semantic representations are identical) and feature only local lexical differences (*enable* vs. *allow* and *frequently* vs. *often*), as well as different pronominalizations (*John's* [income] vs. *his* [income] and [enables] *him* vs. [allows] *John*).

Sentences (4c-d) are approximate mutual paraphrases, differing slightly in their propositional contents (*thanks to* [expressing 'cause'] vs. *with* [expressing 'means'] and the omission of 'able' in the second sentence), but having the same communicative and stylistic orientation.

As for sentences (4a-b) and (4c-d), they are more remote approximate paraphrases, as they differ not only in some aspects of their propositional meanings but also in their respective communicative organizations, reflected in their globally different syntactic structures.

We could go on with the comparisons, but what has been said seems enough to illustrate the fact that paraphrases can differ more or less substantially with respect to their semantic, communicative and/or stylistic features, ranging from very close to quite distant. Needless to say, the greater the semantic distance between paraphrases, the more interesting and difficult their description becomes.

3 Types of Paraphrastic Links Found in the Corpus and Corresponding Paraphrasing Rules

Three types of paraphrastic links, or paraphrastic equivalences, were found in the corpus: 1) lexical syntactic equivalences, 2) (exact) semantic equivalences and 3) semantic quasi-equivalences. We will take them in turn, along with the corresponding paraphrasing rules.

3.1 Lexical-Syntactic Equivalences

These are the equivalences between lexical items and syntactic constructions within which they appear. In the simplest case, they involve local variation of the propositional content based on synonymic substitutions, some of which are exact and some approximate; cf.:

John's	comfortable income	enables	him	(to) travel	a lot	
his	high earnings	permits	John	(to) go on trips	lots	
	high salary	allows		(to) take trips	frequently	
	comfortable returns	lets		(to) do trips	often	
	adequate funds			(to) go travelling	regularly	
2x	5x	4x	2x	5x	5	= 2 000

Table 4: Some lexical-syntactic equivalences found in our corpus

Note the large number of paraphrases that can be obtained by these relatively simple substitutions, illustrating the high paraphrastic potential of Language.

Paraphrases of this type are modeled by means of well-known lexical-syntactic paraphrasing rules (Žolkovskij & Mel'čuk 1967; Mel'čuk 1974: 141-176, 1992, 2013: 137-197). These rules, formulated in terms of lexical functions (Mel'čuk 1974: 78-109; Wanner, ed., 1996; Mel'čuk & Polguère 2021), operate on dependency-based Deep-Syntactic Structures (DSyntSs) of sentences and are, just like the formalisms in which they are couched, cross-linguistically universal.

Two equivalent DSyntSs and two lexical-syntactic paraphrasing rules that relate them are given in Figs 1 and 2 below.

REMARK. In these DSyntSs and paraphrasing rules, we see the following lexical functions: Oper_1 (a particular light verb), S_0 (a nominalization), $\mathrm{Magn}^{\mathrm{FREQ}}$ (an intensifying collocate bearing on the frequency of occurrence of the fact denoted by the base of the collocation) and Syn (a synonym of a lexical unit).

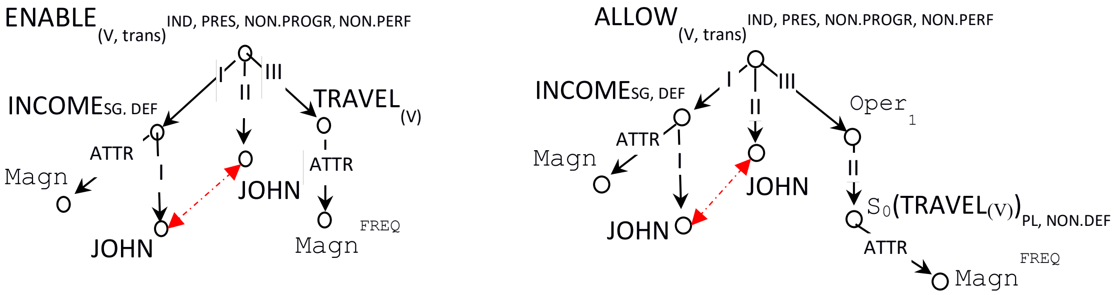
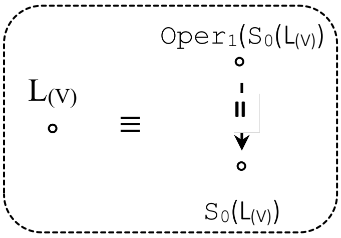

ENABLE (V, trans) IND, PRES, NON.PROGR, NON.PERF
I
II
III
INCOME SG, DEF
TRAVEL (V)
ATTR
Magn
JOHN
ATTR
JOHN
Magn FREQ
ALLOW (V, trans) IND, PRES, NON.PROGR, NON.PERF
I
II
III
INCOME SG, DEF
Oper1
ATTR
Magn
JOHN
II
JOHN
S0(TRAVEL(V)) PL, NON.DEF
ATTR
Magn FREQ

Figure 1: DSyntS of (1a) and an equivalent DSyntS



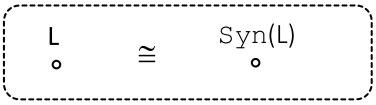


Figure 2a: Light verb fission (Rule$^{\mathrm{EQ\_EX\text{-}SYNT}}$ 1)   Figure 2b: Synonymic substitution (Rule$^{\mathrm{EQ\_EX\text{-}SYNT}}$ 2)

The rule in Fig. 2a allows for the substitution (to) *travel*$_{(\mathrm{L(V)})}$ ~ (to) *take <do, go on>*$_{(\mathrm{Oper1})}$ *trips*$_{(\mathrm{S0(L(V))})}$, and that in Fig. 2b accounts for the substitution (to) *enable*$_{(\mathrm{L})}$ ~ (to) *allow*$_{(\mathrm{Syn(L)})}$.

Meaning-Text paraphrasing system contains some hundred rules of the type illustrated in Fig. 2, capable of treating a wide range of quite sophisticated lexical-syntactic paraphrases.

3.2 Semantic Equivalences

Semantic equivalences fall into two major subtypes. Semantic-propositional equivalences are based on the operation of semantic decomposition, allowing for a description of a given non-elementary lexical meaning ‘s’ in terms of the meanings simpler than ‘s’.[2] In our approach, these equivalences are modeled by means of semantic expansion/reduction rules—actually, (part of) lexicographic definitions of corresponding lexical units (see Fig. 4 below). These rules operate on semantic structures [SemSs] of sentences and are needed to discover semantic paraphrastic links, not accessible at the deep-syntactic level of representation. They are language specific (depending on the available lexical stock), but their formal type is cross-linguistically universal.

Semantic-communicative equivalences hold between configurations of communicative markers, i.e., specific distributions of values of communicative oppositions such as Thematicity, Givenness, Focalization, etc. They are modeled by means of semantic-communicative restructuring rules, relatively new and less widely known than the decomposition rules (see, for instance, Milićević 2007a: 231-245).

Sentence (4c) is an approximate paraphrase of sentence (1a), differing from it both in the propositional content and communicative orientation. Let us demonstrate how our semantic paraphrasing rules can be used to produce the former from the latter. The underlying representations of (1a) and (4c) follow:

[2] A meaning ‘s_1’ is simpler than the meaning ‘s_2’ if ‘s_1’ can be used within the decomposition of ‘s_2’ and the converse does not hold. Thus, ‘look’ is simpler than ‘stare’ since ‘stare’ = ‘look in a particular way’ and ‘look’ ≠ ‘stare in a particular way.’

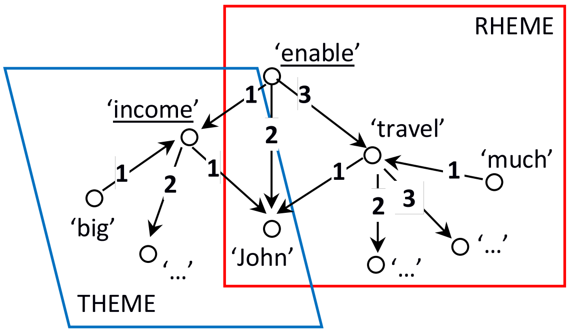


Figure 3a: SemS and Sems-CommS of (1a)

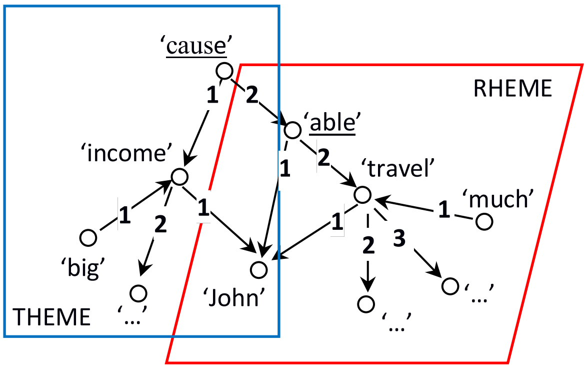


Figure 3b: SemS and Sems-CommS of (4c)

We start by decomposing the semanteme ‘enable’ in the SemS of (1a), using (the equivalence part of) the rule indicated in Fig. 4 below: ‘income$_X$ enables John$_Y$ to travel$_Z$’ = ‘income$_X$ causes that John$_Y$ is able to travel$_Z$.’ This allows us to “extract” the semantemes ‘(to) cause’ and ‘(be) able.’ We proceed to a restructuring of the semantic-communicative structure of (1a), applying to it the rule in Fig. 5, which moves the theme ~ boundary and changes the communicatively dominant semanteme of the semantic rheme from ‘(to) cause$_X$’ (extracted from ‘(to) enable’) to ‘(be) able$_Z$’.[3]

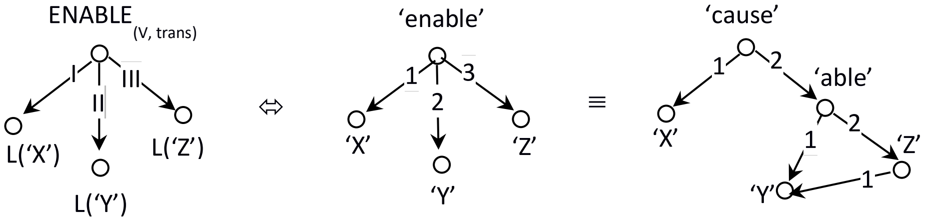


Figure 4: A semantic expansion/reduction rule (using the lexicographic definition of ENABLE$_{(V, trans)}$)

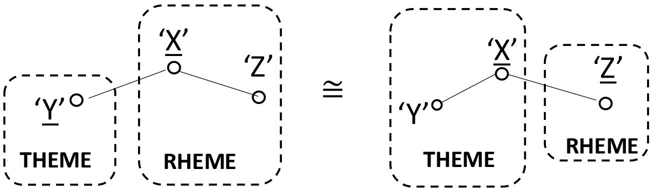


Figure 5: A semantic-communicative reconstruction rule

In the subsequent lexicalization and arborization of the representation underlying sentence (4c), ‘(be) able’ is expressed as the main predication, while ‘to.cause’ is implemented by a prepositional phrase *thanks to* N.

Other implementations of this representation are of course possible; they may involve only alternative expressions of the propositional content or a change of the propositional content itself (requiring the application of additional paraphrasing rules), resulting in more distant paraphrases of (4c); cf.:

thanks to	John’s	comfortable income	he	is able	(to) travel	a lot	
because of	his	high earnings	John	is free	(to) go on trips	lots	
due to		high salary		has the ability	(to) take trips	frequently	
owing to		comfortable returns		has the opportunity	(to) do trips	often	
as a result of		adequate funds		can		regularly	
				can allow himself			
				can afford			
				gets			
				manages			
5x	2x	5x	2x	9x	4x	5	= 18 000

Table 5: Some paraphrases of sentence (4c)

[3] A communicatively dominant semanteme of a Thematic/Rhematic area (defined over a SemS) is the semanteme to which the entire area can be reduced—a sort of a minimal paraphrase of this area.

3.3 Semantic Quasi-Equivalences

These equivalences underly approximate paraphrases such as those in (5) below; the starting sentence (1a) is repeated for convenience as (5a):

(5) a. *John's comfortable income enables him to travel frequently*.
b. *John makes a lot of money; therefore, he is able to travel frequently*.
c. *If John didn't make as much as he does, he wouldn't travel so often*.
d. *John makes enough money to be able to travel a lot*.
e. *With a comfortable income, John goes travelling quite a bit.*
f. *John must make a ton of money: he goes on vacation all the time.*

They are modeled by means of semantic quasi-equivalence rules (Milićević 2007a: 190-230, 2007b, 2021), global substitution rules which manipulate very general meanings, close to semantic primitives (Wierzbicka 1996, 2021), and are most likely universal.

Quasi-equivalence rules necessary to relate sentence (5a) to its paraphrases (5b)-(5f) are indicated in Table 6 below; all but the last one had been previously identified and described within the Meaning-Text approach.

(5a) to (5b)	CAUSE ~ CONSEQUENCE	'P causes Q' ≅ 'P, consequently Q'	✓
(5a) to (5c)	CAUSE ~ CONDITION	'P causes Q' ≅ 'If (not) P, then (not) Q'	✓
(5a) to (5d)	CAUSE ~ SUFFICIENT CONDITION	'P causes Q' ≅ 'P is.sufficient.for Q'	✓
(5a) to (5e)	CAUSE ~ MEANS	'P causes Q' ≅ 'By.means.of P, Q'	✓
(5a) to (5f)	CAUSE ~ CORROBORATION	'P causes Q' ≅ 'P, as.corroborated.by Q'	✗

Table 6: Semantic quasi-equivalences between sentences in (5) and the corresponding paraphrasing rules

The CAUSE ~ CONSEQUENCE rule written in the semantic network formalism is given in Figure 6:

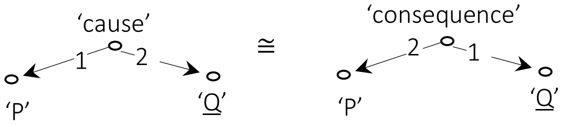


Figure 6: A semantic quasi-equivalence rule (a global substitution)

The application of semantic quasi-equivalence rules invariably requires prior semantic decompositions of meanings present in the initial semantic structure (carried out by semantic expansion/reduction rules introduced in section 2.1) in order to extract the semantic component(s) they manipulate—in our case, the semanteme '(to) cause'. They also regularly trigger major modifications of the communicative structure of the initial sentence (performed by semantic-communicative reconstruction rules like the one in Fig. 5).

Let us sketch the process of production of sentence (5b) starting from the semantic representation underlying sentence (5a), given in a verbal, or textual, form (6a) below. (The semantemes intensifying 'money' and 'travel' are omitted from the representation for simplicity's sake; the communicatively dominant semantemes are underlined).

(6) a. 'John's having$_P$ money <u>enables</u> him to travel$_Q$'
b. 'John's having$_P$ money '<u>causes</u> that he is.able' to travel$_Q$'
c. 'John's being.able to travel$_Q$ <u>is.a.consequnce.of</u> his having$_P$ money'
d. 'John <u>has</u>$_P$ money; as a.consequence, he <u>is.able</u> to travel$_Q$'

Semantic decomposition of 'enable' (by the expansion/reduction rule in Fig. 4 above applied from left to right) results in the representation in (6b). Then the global substitution rule in Fig. 6 is applied to (6b), yielding (6c). Finally, a communicative reconstruction rule (that will not be shown) is applied to (6c) in order to put 'be.able' in the communicatively dominant position with respect to 'consequence', which get us to (6d). A result of this last rule's application is the inversion of subordination (head switching) happening in the subsequent phases of synthesis (at the deep-syntactic level of representation).

The representation in (6c) can of course be implemented by sentences other than (5b), for example the following ones:

(7) a. *John has money; therefore <so, consequently> he can travel.*
b. *John has money and (therefore) can travel.*
c. *John has money; he can travel.*

To wrap up, let me mention two semantic quasi-equivalences underlying some paraphrases in the corpus that are not causation-related; the first one holds between the sentences in (8) and the second one between those in (9); the relevant fragments of sentences are bolded:

(8) a. *Because of his comfortable income, John* ***can travel*** *a lot.*
b. *John* ***travels*** *a lot thanks to his comfortable income.*

(9) a. *Thanks to his remunerative job, John travels* ***often*** *<****frequently****>.*
b. *Travelling has been made* ***easy*** *thanks to John's high salary.*

The corresponding quasi-equivalence rules are indicated in Table 7; the second rule hadn't been identified and described before.

(8a) to (8b)	ABILITY ~ HABIT/FREQUENCY	'X is.able to do P ' ≅ 'X usually/often does P'	✓
(9a) to (9b)	HABIT/FREQUENCY ~ EASE	'X usually/often does P' ≅ 'doing P is.easy for X'	✗

Table 7: Another two semantic quasi-equivalences and the corresponding paraphrasing rules

These quasi-equivalences are not purely linguistic in nature; they exploit some everyday knowledge about the world: in the first case, that being able to do something entails being in the habit of doing this thing, if it turns out beneficial or pleasant for us; in the second, that things we do often and up being easy for us to do. Thus, the paraphrases they allow us to produce are partially pragmatically based.

4 Conclusion

As predicted at the outset of the study, most paraphrasing rules applied (unconsciously) by the speakers that produced the paraphrases in our corpus already exist within the Meaning-Text paraphrasing system. More specifically, semantic (quasi-)equivalence rules and semantic-communicative restructuring rules were exploited for global restructuring of the semantic representation underlying the starting sentence; they were often used in conjunction with lexical-syntactic equivalence rules for more local restructuring.

A few paraphrastic links were found that were not accounted for by the existing paraphrasing rules:

- rhetorical additions (***We know that*** *John does well …* ; *How often John travels* ***tells us*** *that …*);
- «causation ~ corroboration» link, illustrated in (5a)-(5f), also rhetorical in nature;
- «habit ~ frequency/ease» link, as in (9), a semantic equivalence with a hint of pragmatic knowledge.

To find more paraphrastic links not accounted for by Meaning-Text paraphrasing rules, a larger corpus of paraphrases is needed.

Future studies could focus on the description of rhetorical paraphrastic links, like those just mentioned, and pragmatically based paraphrases, such as those illustrated in example (3).

Acknowledgements

I am grateful to Igor Mel'čuk for his remarks on a prefinal version of the paper.

References

Féry, Caroline & Ishihara, Shinchiro, eds. 2016. *The Oxford Handbook of Information Structure*. Oxford: Oxford University Press.

Fuchs, Catherine. 1980. *Paraphrase et théorie du langage. Contributions à une histoire des théories linguistiques contemporaines et à la construction d'une théorie énonciative de la paraphrase*. PhD Thesis. Paris: Université Paris VII.

Marengo, Sébastien, ed. 2021. *La Théorie Sens-Texte. Concepts clefs et applications*. Paris: L'Harmattan.

Martinot, Claire. 2009. Reformulations paraphastiques et stades d'acquisition en français langue maternelle. *Cahiers de Praxématique*, 52: 29-58.

Mel'čuk, Igor. 1974. *Opyt teorii lingvističeskix modelej Smysl-Tekst*. Moscow: Nauka. [Reprinted in 1999, Moscow: Jazyki russkoj kul′tury.]

Mel'čuk, Igor. 1992. Paraphrase et lexique: la théorie Sens-Texte et le Dictionnaire explicatif et combinatoire. In: Mel'čuk, I. *et al*., *Dictionnaire explicatif et combinatoire du français contemporain. Recherches lexico-sémantiques III*. Montréal: Presses de l'Université de Montréal; 9-59.

Mel'čuk, Igor. 2001. *Communicative Organization in Natural Language*. Amsterdam/Philadelphia: Benjamins.

Mel'čuk, Igor. 2013. *Semantics. From Meaning to Text*, vol 2. Amsterdam/Philadelphia: John Benjamins.

Mel'čuk, Igor & Polguère, Alain. 2021. Fonctions lexicales dernier cri. In: Marengo, S., ed., 2021; pp. 75-155.

Milićević, Jasmina. 2007a. *La paraphrase. Modélisation de la paraphrase langagière*. Berne: Peter Lang.

Milićević, Jasmina. 2007b. Semantic Equivalence Rules in Meaning-Text Paraphrasing. In: Wanner, L., ed., *Selected Lexical and Grammatical Issues in Meaning-Text Theory. In Honor of Igor Mel'čuk*. Amsterdam/ Philadelphia: John Benjamins; 267-297.

Milićević, Jasmina. 2021. Modélisation de la paraphrase linguistique approximative. In: Marengo, S., ed, 2021; pp. 51-74.

Quirk, Randolph, Greenbaum, Sidney, Leech, Goeffrey & Svartvik, Jan. 1985. *A Comprehensive Grammar of the English Language*. London: Longman.

Wanner, Leo, ed. 1996. *Lexical Functions in Lexicography and Natural Language Processing*. Amsterdam/Philadelphia: John Benjamins.

Wierzbicka, Anna. 1996. *Semantics, Primes and Universals*. Oxford/New York: Oxford University Press.

Wierzbicka, Anna. 2021. Semantic Primitives, Fifty Year Later. *Russian Journal of Linguistics*, 25/2: 317-342.

Žolkovskij, Aleksandr & Mel'čuk, Igor. 1967. O semantičeskom sinteze. *Problemy kibernetiki*, 19: 177-238. [French translation: Sur la synthèse sémantique. 1970. *TA Informations*, 2: 1-85.]

Appendix: Corpus of Paraphrases

Below we list all eighty paraphrases of the sentence *John's comfortable income enables him to travel a lot*.[4]

2-CLAUSE REALIZATION

P = 'John has money'; Q = 'John can travel'

Main fact: the one implemented as the main predication/matrix clause.

Completive clauses and nominal relative clauses are considered clause elements (parts of clauses).

A Coordination

John has a good salary and can afford frequent travel. | *John must make a ton of money: he goes on vacation all the time.* | *John makes a lot of money; therefore, he is able to travel frequently.* [2]

B Subordination

B1 Main fact: P

John makes good money, which permits him to take a lot of trips. | *John makes a lot of money, which lets him travel often.* | *John has a high salary, which allows him to travel lots.* | *John is paid well, which is why he can go on so many vacations.* | *John receives comfortable wages, which allows him to travel several times a year.* | *John makes enough money that he can travel a lot.* | *We know John does well (for himself) because he travels a lot.* [7]

[4] The paraphrases were produced by eight students of the course FREN 4046 (*Expression écrite*) given at Dalhousie University (Halifax, Canada) in the 2017-2018 academic year.

B2 Main fact: Q

John is able to travel a lot because he has a high income. | John has the ability to travel a lot because he makes a comfortable income. | John travels a lot because he has a comfortable income. | John travels a lot because he makes a decent amount of money. | John often goes on trips because he has a very good income. | John goes on a lot of trips because he's well off. | John travels a lot since he has a comfortable income. | John can travel lots, since he makes a comfortable amount of money. | John travels a lot as he receives a comfortable income. | John has been able to travel on numerous occasions, as he makes a decent salary at his job. | *John can travel a lot as he has an income that is comfortable for him. | If John didn't make as much as he does, he wouldn't travel as often.* [12]

1-CLAUSE REALIZATION

C1 Main fact: causation

John's comfortable returns allow him to travel often. | His comfortable income allows John to travel often. | John's earnings allow him to travel a lot. | John's comfortable income allows him to take lots of excursions. | John's adequate income allows him to do many trips per year. | John's untroubled economic status permits him the luxury of travelling often. | John's livelihood grants him the freedom to take trips regularly. | The wages → pay [John experiences → receives, earns, is paid *give him a lot of travel opportunity. | John's frequent vacations have been enabled by his comfortable salary. | John's ability to travel a lot is made possible by his comfortable income. | John's vacations have been made possible due to the amount of money he makes at his job. | Travelling has been made easy thanks to John's prosperous →* generous, high *salary. | Being able to travel a lot is a result of John's good salary. | It is John's comfortable income that supports his travel habits.* [13]

C2 Main fact: P

John earns enough money to afford to travel often. | John makes enough money to allow him to take frequent vacations. | John makes enough money to be able to travel a lot. | John's revenue → income, earnings *is/*are *such that he can travel often without economic →* financial *pressure. | The money [that John makes] is sufficient to pay for all the trips he takes. | The income [that John makes] is comfortable enough to allow him to travel a lot. | [How often John travels] tells us he makes a good living. | The reason why John is able to take so many trips a year is his great revenue.* [8]

C3 Main fact: Q

John, enabled by a comfortable income, travels a lot. | John is able to travel often because of his comfortable income. | John is able to travel several times during the year because of his comfortable wages. | John can travel often because of his earnings. | John can take many trips thanks to his high-paying job. | John can go on many vacations thanks to his comfortable income. | John can travel a lot thanks to his comfortable salary. | John often gets to travel thanks to his income. | John gets to travel frequently, thanks to his high-paying job. | John travels a lot, thanks to his comfortable income. | John travels frequently thanks to his comfortable salary. | Thanks to John's comfortable income, he can travel a lot. | Thanks to his remunerative job, John travels often. | Thanks to John's substantial income, he is able to travel often. | Thanks to his comfortable income, John can travel often. | Thanks to his comfortable income, John is able to travel a lot. | Due to the fact that John has a nice income, he goes on many trips. | Due to a generous salary, John travels often. | Because of his comfortable income, John can travel a lot. | Because John makes a lot of money, he has been able to see a lot of the world. | Because John has a job that earns him a prosperous income → a lot of money*, he can travel frequently. | Because John earns a good-sized →* good, high *salary, he has been able to visit many places. | Because of his good salary, John can travel often. | As a result of his high earnings, John gets to see the world. | It is because John has a comfortable income that he can travel as much as he does. | Earning quite a bit of money, John travels often. | Making enough money, John travels a lot. | Making a comfortable amount of money at work, John can travel several times during the year. | Receiving a high income, John can travel a lot. | Having a comfortable income, John is able to travel a lot. | Having a considerable amount of cash, John can travel a lot. | With a comfortable income, John goes travelling quite a bit. | With the money John makes, he is able to travel a lot. | Without John's income, he would never be able to afford his numerous trips. | Without the adequate funds that John has thanks to his job, he would not be able to take as many trips. | Travelling is something John can afford to do due to his substantial salary. | John's frequency of travel is correlated to his high remuneration.* [37]

Is one head enough?
Mention heads in coreference annotations compared with UD-style heads

Anna Nedoluzhko
Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
nedoluzhko@ufal.mff.cuni.cz

Michal Novák
Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
mnovak@ufal.mff.cuni.cz

Martin Popel
Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
popel@ufal.mff.cuni.cz

Zdeněk Žabokrtský
Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
zabokrtsky@ufal.mff.cuni.cz

Daniel Zeman
Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
zeman@ufal.mff.cuni.cz

Abstract

We present an empirical study that compares mention heads as annotated manually in four coreference datasets (for Dutch, English, Polish, and Russian) on one hand, with heads induced from dependency trees parsed automatically, on the other hand. For parsing, we used UDPipe 2.6, a modern parser trained using the Universal Dependencies collection. We show that majority of mismatches (64%–94%) can be attributed to several classes of systematic differences in how the notion of head is treated in the respective data resources, while mismatches caused by parsing errors are relatively rare (4%–15%). Our conclusion is that consistency would be gained in (and across) coreference resources after migration to UD-style mention heads, without losing substantial information. This can be achieved with sufficient accuracy using modern dependency parsers even for coreference corpora that lack manual head annotation.

1 Introduction

Coreference is a relation between expressions in a text which refer to the same real-word entity or event; the referring expressions are called *mentions*. In most datasets annotated with coreference relations (see Nedoluzhko et al. (2021) for a survey), a mention is represented simply by specifying the corresponding sequence of tokens (called a mention *span*), typically contiguous, mostly belonging to a single sentence.

Naturally, a mention span can be analyzed syntactically. There is a vague consensus that some tokens, often delimited syntactically, carry more important information from the coreference resolution perspective than other tokens. The most crucial part is called a *minimum span* by some (as opposed to *maximum span* denoting the whole span; see e.g. Uryupina et al. (2020), Hirschman and Chinchor (1998)), or simply a *head* by others (Ogrodniczuk et al., 2013), which is the term we adhere to. Identifying a mention's head is motivated not only linguistically, but also technically: with a long-span mention, there is a higher risk of annotation noise and requiring the exact match when evaluating span boundary prediction could be misleading. See e.g. Uryupina et al. (2020), Elsner and Charniak (2010), Peng et al. (2015), or Wiseman et al. (2016) for more arguments on the importance of head for the task of coreference resolution.

The notion of mention head in coreference annotations largely resembles the notion of head in dependency treebanks; however, with a few exceptions such as the Prague Dependency Treebank (Hajič et al.,

2020) (PDT for short), the coreference and dependency-treebanking annotation efforts remain isolated to a surprising degree.

In this paper we present a novel empirical study that compares manually annotated mention heads within coreference annotation projects with syntactic heads identified automatically by a modern parser trained on dependency treebanks from the Universal Dependencies (UD) collection (de Marneffe et al., 2021). Our long-term motivation is based on the expectation that making the mention head notion convergent with heads induced from dependency structures following the UD guidelines could result in (a) improved annotation consistency in existing coreference datasets, and (b) more efficient and faster development of new coreference datasets (e.g. because of possible reuse of UD-related software tools), especially when it comes to extensions to multiple languages. However, in a shorter-term perspective, we should first try to explain the nature of differences between mention heads as annotated in existing coreference datasets on the one hand and UD-compliant heads of mentions on the other.

We make use of the CorefUD 0.2 collection, which contains 17 coreference datasets for 11 languages converted to a common annotation scheme (Nedoluzhko et al., 2021). There is some notion of head used explicitly or implicitly in 13 out of the 17 datasets. However, we limit ourselves only to datasets in which mention heads are marked explicitly, and, at the same time, whose coreference annotations were created without using full-fledged hand-annotated syntactic structures (dependency or constituency). Thus, for example, the Prague Dependency Treebank dataset is excluded, since coreference and dependency annotations are tightly connected in it by design. The selection criterion leads to four resources: ARRAU (Uryupina et al., 2020) for English, COREA for Dutch (Hendrickx et al., 2008), Polish Coreference Corpus (Ogrodniczuk et al., 2015; Ogrodniczuk et al., 2013), and Russian Coreference Corpus (Toldova et al., 2014). Datasets in CorefUD 0.2 have been parsed using the UDPipe 2 tool (Straka, 2018) with very recent parser models.

The rest of the paper is structured as follows. Section 2 summarizes different approaches to the notion of head, both from the syntactic and coreference perspectives. Section 3 gives basic information about the four coreference data resources included in our study. Section 4 describes our annotation of mentions selected from the four resources; we focused on mentions in which the mention head marked in the original coreference resource does not match the root of the mention in terms of automatically parsed UD tree. Section 5 analyzes and exemplifies types of such mismatches. Finally, Section 6 concludes.

2 Related work

2.1 Head in dependency annotation schemes

One can easily foresee that – in spite of the recent progress in parsing technology – there will be non-negligible amount of head mismatches which are due to parsing errors; similarly, a non-zero amount of errors in manual annotation of mention heads can be expected too. However, we are interested in more principled sources of variability of the notion of head.

It was recognized by dependency-oriented scholars long time ago that multiple types of dependencies may be distinguished (especially syntactic and semantic ones), and that syntactic dependencies should not be confused with other types of relations, see e.g. a discussion on “double dependency” and “mutual dependency” in Mel’čuk and others (1988). However, the trend in the current dependency treebanking in the last decade is inclined rather to maximize simplicity and robustness, with Universal Dependencies (de Marneffe et al., 2021) being the most prominent representative, rather than to design multilayered annotation schemes with strictly separated hypotactic and paratactic “brackets” (with the latter ones possibly interpreted as additional “dimensions” of dependency trees (Sgall, 1998)) on each layer. This trend has a clear rationale especially if quick portability to multiple languages is one of the modern priorities, however, on the other hand, such formally simple structures are prone to various confusions concerning the notion of head.

The fact that in some cases there is no unique obvious way for choosing a head of an expression, has been noticed many times as it has inevitable practical consequences in dependency-oriented projects. Above all, annotators’ intuition concerning the dependency structure of sentences is insufficient for reaching reasonable annotation consistency, and thus artificial annotation rules must be introduced by convention. This can be illustrated by extensive annotation guidelines developed basically in every mature dependency treebanking project. We believe that most of the observed variability in the notion of head can be attributed to the following sources, as discussed in more detail the subsections below:

- opposite direction of syntactic and semantic dependencies (and other non-parallelisms),
- representing functional words as nodes of their own,
- representing paratactic relations within dependency trees,
- no obvious head-dependent asymmetry in a syntactic constituent.

2.1.1 Opposite direction of syntactic and semantic dependencies

Several types of constructions are recognized in literature in which the direction of a syntactic dependency relation manifested by overt surface morphosyntactic means (such as agreement) is opposite to what is considered as semantic dependency; the syntactic and semantic heads are swapped, in other words.

When designing treebank annotation guidelines, the authors either have to indicate whether syntactic or semantic dependencies are the preferred ones, or, alternatively, provide technical means for capturing both. The latter option can be illustrated by PDT, in which there are two separate dependency trees, one of them capturing surface syntax and the other one capturing deep syntax and semantics (to some extent). Similarly, the Enhanced representation in Universal Dependencies (Nivre et al., 2020, Section 3.4) adds extra edges to make explicit some semantically relevant relations that are otherwise implicit in basic dependencies.

2.1.2 Functional words

If functional words have nodes of their own in a dependency representation, it can lead to problems related to head choice. A functional word is usually clearly associated with an autosemantic (meaningful) word, however, it is not clear which of them should be the head (more precisely, either choice can be justified with reasonable arguments, and one simply has to choose). Examples of such pairs are a preposition and a noun in a prepositional group, an auxiliary verb and an autosemantic verb in a complex verb form, or a determiner and a noun. For instance, if a prepositional group is considered, PDT surface syntax guidelines make the preposition the governor and put the noun below, while the two are connected the other way round in UD. If an auxiliary verb in a complex verb form bears congruent categories, then it becomes the governor in the PDT, while the autosemantic component of the complex verb form is the governor in most cases. Both PDT and UD annotation styles attach determiners below nouns being determined, but determiners are treated as governors of noun phrases in the Danish Dependency Treebank (Kromann et al., 2003).

A more complex example is that of expletives: in some cases insertion of expletive expressions (such as pronouns) is needed or preferred in a language, for instance if valency of a matrix clause verb requires a morphological case to be manifested with its argument, but the argument is a subordinating clause. Then, again, it is not clear whether the expletive pronoun or the subordinating clause head should be chosen as the head of it all.

2.1.3 Paratactic structures

In the case of parataxis, two syntactically connected expressions are in an equal relation with each other, instead of being subordinated one to the other. In other words, there is no head-dependent asymmetry. Typical examples are coordination and apposition constructions. Especially coordination has always been a nightmare for dependency grammarians, as it is very frequent and interferes in various ways with dependency relations. However, as long as we preserve the design decision that all we have for syntactic representation is nodes and edges, we have to encode paratactic constructions in this way too. There is a surprising number of different possible encodings for doing so, and a smaller, but even more surprising number of encodings that has been really used in existing treebanks, see Popel et al. (2013) for a survey. However, in most cases it boils down to either using coordination conjunction as the head node, or using one of the conjuncts as the head, selected in some canonical way.

2.1.4 No overt head-dependent asymmetry

Besides paratactic structures, there are also other types of expressions in which we perceive some internal structure and for which we do not possess intuition about what should be the head, but which are not paratactic either. A frequent example is a personal name consisting of a given name and a family name. UD has a dedicated relation type, `flat`, which is used in such exocentric constituents; the first word serves as the technical head, but there is no claim that it is a syntactically (or semantically) motivated head.

To summarize, head choice is far from obvious in various cases, which has both deeply linguistic and purely technical reasons; such situations can only be resolved unambiguously by adhering to artificial annotation rules.

2.2 Head in coreference annotation schemes

For a better orientation, we suggest to classify language data resources containing coreference annotation tentatively as follows:

- head-agnostic approaches,
- head-aware approaches,
- head-centric approaches.

2.2.1 Head-agnostic approaches.

In head-agnostic approaches, a mention is considered the only meaningful unit that is needed for annotating coreference relations and no attempt to find its internal structure is made (at least not to our knowledge).

Examples of head-agnostic approaches are Potsdam Commentary Corpus (Bourgonje and Stede, 2020), the English-German parallel coreference corpus ParCorFull (Lapshinova-Koltunski et al., 2018), and Lithuanian Coreference Corpus (Žitkus and Butkienė, 2018).

2.2.2 Head-aware approaches.

In head-aware approaches, a mention delimited as a sequence of tokens is still the main entity, however, its internal structure is analyzed syntactically[1] (completely or partially) and/or its head is marked explicitly.

Examples of head-aware approaches are Spanish and Catalan data contained in AnCora (Recasens and Martí, 2010) and English data contained in ARRAU (Uryupina et al., 2020).

2.2.3 Head-centric approaches.

In head-centric approaches, it is the head of a mention that is considered to be the argument of a coreference relation, while the exact span in terms of a token sequence is less important (or even left underspecified). Coreference datasets from the PDT family, in which coreference relations connect tectogrammatical (deep-syntactic) nodes and mention span is defined only implicitly, are examples of this approach.

Examples of head-centric approaches are the Prague Dependency Treebank (Hajič et al., 2020) and the Prague Czech-English Dependency Treebank (Nedoluzhko et al., 2016).

3 Coreference datasets with hand-annotated mention heads

Our analysis is based on four datasets from CorefUD 0.2[2] whose original source corpora contain manual annotation of mentions: ARRAU, Polish Coreference Corpus, COREA, and Russian Coreference Corpus.

3.1 ARRAU

The ARRAU Corpus of Anaphoric Information (Uryupina et al., 2020) (further abbreviated as English-ARRAU) is a multi-genre corpus of English which provides large-scale annotations of a wide range of anaphoric phenomena. In English-ARRAU, the special attribute MIN (or minimal span) is manually annotated, similarly as it was once decided for MUC-7 (Hirschman and Chinchor, 1998). This attribute

[1] An analysis whether or not coreference mentions do correspond to subtrees of UD trees can be found in Popel et al. (2021), without a special attention paid to heads, though.

[2] `http://hdl.handle.net/11234/1-4598`

corresponds to the head noun for non-proper nominal mentions, or to the entire proper name (for example, first name and surname) in case of multi-word named entities. It is not explicitly stated in the guidelines, if syntactic or semantic heads are preferred. According to the MUC-7 coreference task definition[3], it maybe deduced that syntactic heads are preferred. However, this has not been stated explicitly for MUC-7 neither.

3.2 Polish Coreference Corpus

The Polish Coreference Corpus (Ogrodniczuk et al., 2013; Ogrodniczuk et al., 2015) (further abbreviated as Polish-PCC) is a corpus of Polish nominal coreference built upon the National Corpus of Polish. In Polish-PCC, semantic heads, i.e. the most important words from the point of view of the mention's sense, are annotated. The semantic head of a typical nominal group corresponds to the syntactic head but there are some exceptions. For example, in numeral groups like *duzo pieniedzy* 'a lot of money', or *trzech z was* 'three of you', the numeral is the syntactic head, and the noun is the semantic head and is annotated as such in Polish-PCC. The reason for such decision is the claim that coreference is a phenomenon on the level of semantics and discourse more than on the syntactic level. Thus, understanding the semantically central elements should help establish discourse links. Although not explicitly found in the guidelines, the head is understood semantically (an item with larger semantic weight is annotated as head) also in other types of constructions (*od 1999 roku* 'from the year 1999' with the numeral as a head, *pan Ziolkowski* 'Mr. Ziolkowski' with the surname as a head, etc.).

3.3 COREA

The COREA coreference corpus (Hendrickx et al., 2008) (further abbreviated as Dutch-COREA) is a collection of written and transcribed oral texts in Dutch annotated for creating a coreference resolution system. Mentions are strings of text with specially distinguished heads which are defined as minimum strings representing semantic heads of the constituents. Nevertheless, rather than annotated from scratch, the semantic heads were acquired by manual post-editing of the heads obtained from syntactic representation of the underlying texts. For multi-word named entities, the head includes all words of the corresponding entity.

3.4 Russian Coreference Corpus

Russian Coreference Corpus (Toldova et al., 2014) (further abbreviated as Russian-RuCor) is annotated with anaphoric and coreferential relations between noun groups. Mentions are annotated as linear spans, with additionally distinguished heads. Similarly as for English-ARRAU and Dutch-COREA, heads are defined as one-word syntactic heads for common nouns and as sequences of words for multi-word proper nouns. For 'common noun + proper noun' constructions like *the Pushkin street*, the guidelines require the whole multi-word sequences to be annotated as heads, but in the annotated data, only one word is chosen as head (mostly the proper noun).

The comparison of the guidelines for head annotation in the resources under analysis shows that there are differences in the following aspects:

- *Syntactic or semantic understanding of heads*: Semantic heads are explicitly claimed to be annotated in Polish-PCC and partly in Dutch-COREA; in English-ARRAU and Russian-RuCor, there is no explicit claim about the syntactic nature of annotated heads but it may be deduced from the guidelines examples;
- *Possibility to annotate multi-word entities as a head*: Possible for multi-word named entities in English-ARRAU, Russian-RuCor and Dutch-COREA and not applied in Polish-PCC;
- *Choice of the head in 'common noun + proper noun' constructions*: the proper name in English-ARRAU, Dutch-COREA and Polish-PCC and both entities in Russian-RuCor;

[3] https://www-nlpir.nist.gov/related_projects/muc/proceedings/co_task.html

	count	[%]				
				annotated head		
CorefUD dataset	all	one-word	non-catena	missing	same	different
Dutch-COREA	26,476	38.9	2.7	4.6	47.2	6.6
English-ARRAU	57,681	30.0	5.4	3.1	56.3	5.3
Polish-PCC	150,706	49.1	5.0	0.1	44.3	1.5
Russian-RuCor	12,632	68.9	1.1	0.1	27.3	2.5

Table 1: Statistics on mentions in the whole dataset. *all* is the total number of mentions in the train section of a given dataset. The other columns show percentage breakdown into mention types described in the first paragraph of Section 4. The types are detected automatically in a given order, so e.g. a non-catena mention with no annotated head is assigned the non-catena type (not missing head). The last column shows a percentage of multi-word catena mentions with a mismatch in annotated and syntactic head; a sample of 100 mentions of this type was annotated as shown in Table 2.

- Different technical conventions for apposition and coordination structures, special construction with dollar, percent, etc.

Dependency trees in the four datasets under discussion have been obtained for CorefUD 0.2 using UDPipe 2 and models trained on UD 2.6, namely on English-GUM (Zeldes, 2017), Polish-LFG (Patejuk and Przepiórkowski, 2018), Dutch-LassySmall (Bouma and van Noord, 2017), and Russian-SynTagRus (Droganova et al., 2018).

4 Annotation of head mismatches

In our study, we focus on mentions, in which the head coming from the original annotation differs from the head of the mention with respect to the tree produced by automatic dependency parsing.[4] Consequently, all one-word mentions are excluded. In addition, we take into consideration only such mentions whose inner dependency structure forms a *catena*, i.e. a connected subgraph of a dependency tree (Osborne et al., 2012).[5] Moreover, we focus only on mentions where at least one head was annotated.[6]

We randomly sampled 100 such mentions from a train section of each of the four CorefUD datasets under analysis. The examples were examined and annotated by the authors of this work. As none of us is a speaker of Dutch, we utilized public machine-translation services in order to understand the example sentences.

During the annotation process, we settled upon the following categories of head mismatches:

- Wrong – we consider the mismatch to be an error.
    - WrongTree – the automatically parsed UD tree is wrong
    - WrongSpan – wrong syntactic head caused by a wrong mention span, usually due to extra tokens.
    - WrongHead – the manual annotation of head is wrong, i.e. it does not follow the original project annotation guidelines (or at least we were not able to find any guideline which would support such head annotation).

[4] Multiple words could be annotated as heads (or minimal span) in the original annotation. In such cases, we focus on mentions where the syntactic head is not among the set of annotated heads.

[5] Note that catena differs from a **subtree**, which is a catena that spans the head and **all** its descendants. Non-catena mentions have multiple nodes that can be considered syntactic heads of the mention (i.e. their dependency parent is not part of the mention).

[6] See Table 1 for statistics on the total count of mentions and their breakdown into the abovementioned types excluded from the annotation (one-word, non-catena, missing-head, same-head).

	OK				WRONG		
CorefUD dataset	COORD	FLAT	NUM	OTHER	TREE	SPAN	HEAD
Dutch-COREA	25	31	11	7	7	7	16
English-ARRAU	1	44	14	13	4	0	25
Polish-PCC	11	21	23	9	15	1	13
Russian-RuCor	0	85	7	2	5	0	1

Table 2: Result of our annotation of differences in annotated and syntactic heads in a sample of 100 mentions in each dataset. Disclaimer: Individual cases of WRONGHEAD may turn out to be cases of OK or vice versa. A deeper analysis of such cases is a subject of future studies.

- OK – the mismatch in head annotations is correct, as the respective guidelines do not agree on the head for a given phenomenon
  - OK-COORD – the first conjunct of a coordination is always marked as a head in UD. The original annotation marks the coordination conjunction or another conjunct as a head, instead. This is an example of the parataxis (see Section 2.1.3).
  - OK-FLAT – UD chooses the first token as head in flat structures (such as names, marked with deprel `flat`), appositions (marked with deprel `appos`) and lists, while the original dataset annotators decided to analyze it as a non-flat structure. This is an example of non-overt head-dependent asymmetry as we describe it in Section 2.1.4, and parataxis for apposition (see Section 2.1.3).[7]
  - OK-NUM – the mismatch is caused by an opposite direction of syntactic and semantic dependencies (see Section 2.1.1). This most often includes numerals and containers (e.g. *a group of people*).
  - OK-OTHER – another subtype of OK.

5 Analysis and discussion

Table 2 summarizes the head mismatches annotation in the selected datasets. As we can see, there is a relatively low number of mismatches caused by wrong parsing. With a slightly larger number of such cases in Polish-PCC, there are just up to 7% of wrongly parsed annotated mentions in English-ARRAU, Dutch-COREA and Russian-RuCor. One of the reasons is that we included only multi-word catena structures into the analysis.[8] The remaining cases of wrong parsing are specific syntactic or derivation constructions, e.g. the deadjectival noun *teściowa /mother-in-law/* in Example 1[9] from Polish which is falsely recognized as an adjective in UDPipe and thus gets a dependent position in the parsed tree. The surprisingly low overall number of parse errors can be justified by comparative simplicity of parsing of noun phrases (the majority of mentions are noun phrases).

[7] Even though appositions are paractic constructions, we rather included them in the OK-FLAT category. The reason is that they are closely related to hypotactic constructions such as *president Trump*, which are in fact treated as appositions in some of the datasets (e.g. Dutch-COREA).

[8] Note that Polish-PCC has the lowest percentage of head mismatches according to the last column in Table 1. Thus, in Polish-PCC we could expect 1.5% · 15% = 0.23% mentions with head mismatch caused by wrong parsing in the whole dataset, while in Dutch-COREA it is twice as much: 6.6% · 7% = 0.46%.

[9] Examples in this work are presented in both glosses and trees. The first line of the gloss shows the original sentence / excerpt / phrase, optionally followed by its word-to-word translation and smooth translation to English. Nodes in the dependency tree show the word form, part-of-speech tag and dependency relation to the node's parent. While in gloss the annotated mention is typeset in bold, `**token**` is used to mark each token of the mention in the tree. The annotated mention head and syntactic head given by the parser are labelled only in the tree.

(1)

Przypomniał	*sobie*	*pazury*	***teściowej***	***doktora***	***Goldmanna***.
He remembered	himself	claws	of mother-in-law	of dr.	Goldmann.

‘He remembered Dr. Goldmann’s mother-in-law’s claws.’

```
─┮ Przypomniał VERB root
 ┝━╸ sobie PRON iobj
 ┝━┮ pazury NOUN obj
 │ │ ┌─╸ **teściowej ADJ amod <======= annot. head**
 │ └─┾ **doktora NOUN nmod:poss <----- synt. head**
 │   └─╸ **Goldmanna PROPN flat**
 └─╸ . PUNCT punct
```

Another apparent general observation is a disproportion of incorrectly parsed or annotated sentences (WRONG labels) between Russian-RuCor (6%) and the other datasets (28–30%). This is likely a consequence of annotation mismatches in proper nouns that prevail in the selected sample (see Section 5.2).

Our analysis reveals a number of mismatches (i) between syntactic heads generated by the UD parser and manually annotated heads in the datasets, but also inconsistencies (ii) across the datasets and (iii) within the annotated datasets themselves. The most typical categories of mismatches (OK-COORD, OK-FLAT and OK-NUM) and annotation inconsistencies are addressed in the following subsections.

5.1 Heads in coordinations

The prevailing reason for mismatches of the OK-COORD type is that the coordination conjunction is annotated as the head of a coordination mention. In total it accounts for 73%.

In cases where a non-first conjunct is annotated as a mention head, the conjunct often comprises information that is shared among all conjuncts, e.g. in Example 2 from Dutch-COREA.

(2)

gezonde	***bacterie-***	***of***	***virusdragers***
healthy	bacteria	or	virus carriers

‘healthy bacteria or virus carriers’

```
 ┌─╸ **gezonde VERB amod**
─┾ **bacterie- X obl <------------- synt. head**
 │ ┌─╸ **of CCONJ cc**
 └─┶ **virusdragers NOUN conj <==== annot. head**
```

5.2 Heads in expressions with proper names

Constructions with proper names form a great deal of the OK-FLAT category. There are different annotation conventions for annotating heads in such constructions in UD (de Marneffe et al., 2021) and across the annotated datasets (see descriptions of the datasets in Section 3). Generally, whereas in phrases like *President Carter* the annotator more often chooses the proper noun as head (because it is referentially concrete), it is the first word (i.e. the general name *President* in our example) according to the UD convention (see Example 3 from English-ARRAU).

(3) *Mr. Hastings was appointed to the federal bench by* ***President Carter***

```
 ┌─┮ Mr. PROPN nsubj:pass
 │ └─╸ Hastings PROPN flat
 ┢─╸ was AUX aux:pass
─┾ appointed VERB root
 │ ┌─╸ to ADP case
 │ ┢─╸ the DET det
 │ ┢─╸ federal ADJ amod
 ┡─┶ bench NOUN obl
 │ ┌─╸ by ADP case
 └─┾ **President PROPN obl <------ synt. head**
   └─╸ **Carter PROPN flat <====== annot. head**
```

Interestingly, there are 85% such cases in the Russian sample, although the guidelines rather advise to label both expressions as a multi-word head. See the expression *журнала Time /Time magazine/* in Example 4, where the annotated mention head is the proper name and the UD head is the common noun *журнала /magazine/*. This type of mismatches is also frequent in other datasets, see e.g. *moja babcia Zofia /my grandma Zofia/* in Example 5 (Polish-PCC) or *vitamine C* in Example 6 (Dutch-COREA).

There is also a non-negligible number of inconsistencies in annotation of multi-word named entities within the datasets. Although, the guidelines require to mark the entire multi-word units as heads in all datasets except Polish-PCC, in some cases, only one more semantically significant word is annotated. See the annotation of only surname in the multi-word name Hillary Clinton in Example 7 from Russian-RuCor.

(4)

Подписка	*на*	*«планшетную»*	*версию*	***журнала***	***Time***
Subscription	to	"tablet"	version	of magazine	Time

'Subscription to the "tablet" version of Time magazine'

```
──┬ Подписка NOUN nsubj
  │ ┌── на ADP case
  │ │ ┌── « PUNCT punct
  │ ├─┼ планшетную ADJ amod
  │ │ └── » PUNCT punct
  └─┼ версию NOUN nmod
    └─┬ **журнала NOUN nmod <----------- synt. head**
      └── **Time PROPN flat:foreign <=== annot. head**
```

(5)

moja	***babcia***	***Zofia***	***Gołąbowa***	*mieszkała*	*w*	*kamienicy*
my	grandma	Zofia	Gołąbowa	lived	in	tenement house

'my grandmother, Zofia Gołąbowa, lived in a tenement house'

```
    ┌── **moja DET det**
  ┌─┼ **babcia NOUN nsubj <------ synt. head**
  │ ├── **Zofia PROPN flat <===== annot. head**
  │ └── **Gołąbowa PROPN flat**
──┼ mieszkała VERB root
  │ ┌── w ADP case
  └─┴ kamienicy NOUN obl
```

(6)

door	***vitamine***	***C***	*of*	*amandelzuur*
by	vitamin	C	or	mandelic acid

'by vitamin C or mandelic acid'

```
  ┌── door ADP case
──┼ **vitamine NOUN nmod <---- synt. head**
  ├── **C PROPN appos <======= annot. head**
  │ ┌── of CCONJ cc
  └─┴ amandelzuur NOUN conj
```

(7)

Запрос	*был*	*подан*	***госсекретарю***	***США***	***Хиллари***	***Клинтон***
request	was	submitted	secretary	USA	Hillary	Clinton

'The request was submitted to the US Secretary Hillary Clinton'

```
  ┌── запрос NOUN nsubj:pass
  ├── был AUX aux:pass
──┼ подан VERB root
  ├─┬ **госсекретарю NOUN iobj <--------- synt. head**
  │ ├── **США PROPN nmod**
  │ └─┬ **Хиллари PROPN appos**
  │   └── **Клинтон PROPN flat:name <==== annot. head**
  └── . PUNCT punct
```

5.3 Heads in expressions with numerals and quantifiers

Head mismatches in constructions with numerals grouped under the OK-NUM category may be further divided into the following subgroups.

Cardinal numerals. Numeral mentions where a noun-like word is modified by a number (e.g. *five cars*) are typical cases of head mismatches. In most of them, the modified word is in fact a currency's name or symbol (e.g. *$25 million*, *vijfhonderd zestig miljoen gulden /five hundred and sixty million guilders/*, *90 млрд рублей /90 billion rubles/*).

Heads are annotated inconsistently across datasets: numerals prevailingly serve as heads in English-ARRAU (Examples 8–9) and Polish-PCC (Example 10), while it is the modified words in Russian-RuCor (Example 13). Nevertheless, annotation of mention heads seems to be inconsistent also within some of the datasets. Let us look into mentions with a *$* symbol as their syntactic head (*$* mention) in English-ARRAU. Out of 727 such mentions scattered over 179 original documents, only in 43% of them the annotated head (minimal span) matches the syntactic head. Interestingly though, if an original document contains a matching *$* mention, on average more than 92% of all *$* mentions in the document are matching, too. The observed inconsistency thus occurs rather across than within original documents, suggesting that it is an artifact of the annotation workload having been distributed among multiple annotators on the document level.

The mismatches between syntactic and annotated heads partly result also from inconsistencies in parses. However, we do not categorize them as parsing errors (WRONGTREE) since UDPipe models almost perfectly mimic the inconsistency that can be seen already across the manually annotated UD subcorpora they were trained on. While syntactic annotation of single-token numerals (e.g. *five cars*) seem to be identical across the languages, it differs considerably for multi-token numerals with large number names such as thousands, millions etc. (cf. Examples 8-13). Moreover, in Russian-SynTagRus the tree of multi-token numerals is shaped differently based on whether the word representing the large number name is in singular (Example 12) or plural (Example 13).

(8) ***3.5 million ounces***

```
   ┌──╼ **3.5 NUM compound**
 ┌─┶ **million NUM nummod**
 ┶ **ounces NOUN root <===== annot./synt. head**
```

(9) ***$25 million***

```
─┮ **$ SYM appos <------------ synt. head**
 │ ┌──╼ **25 NUM compound**
 └─┶ **million NUM nummod <==== annot. head**
```

(10)

40	*milionów*	*złotych*
40	million	złoty

```
 ┌─┮ **40 NUM nummod**
 │ └──╼ **milionów NOUN flat <==== annot. head**
 ┶ **złotych NOUN nmod:poss <---- synt. head**
```

(11)

vijfhonderd	*zestig*	*miljoen*	*gulden*
five hundred	sixty	million	guilders

```
  ┌──╼ **vijfhonderd NUM nummod**
  ┾──╼ **zestig NUM nummod**
─┾ **miljoen NOUN nmod <----- synt. head**
  └──╼ **gulden NOUN nmod <==== annot. head**
```

(12)

две	*тысячи*	*предложений*
two	thousand.GEN.SG	sentences

```
   ┌──╼ **две NUM nummod:gov**
 ┌─┶ **тысячи NUM nummod:gov**
 ┶ **предложений NOUN obl <---- synt. head**
```

(13)

5	*тысяч*	*военных*
5	thousand.GEN.PL	soldiers

```
  ┌──╼ **5 NUM nummod**
─┾ **тысяч NOUN nsubj:pass <-- synt. head**
  └──╼ **военных NOUN nmod <==== annot. head**
```

Syntactically governing numerals and containers. In constructions with governing numerals (e.g. *one of the candidates*, *all of this*) and so-called 'containers' (e.g. *group of tourists*), UDPipe systematically marks the numerals or containers as heads. On the other hand, manual annotation often chooses their syntactic dependent members as more important, putting the emphasis on the semantic point of view. Nevertheless, Examples 14–15 from Polish-PCC and Examples 16–17 from Russian-RuCor show that the manual annotation of mention heads in constructions with containers and governing numerals, respectively, is not systematic. Although we admit there may be another aspect (e.g. semantic salience) that convinced the annotators to label heads in these examples differently, it is neither obvious nor described in the guidelines.

(14)

64	*proc.*	*przemysłu*	*chemicznego*
64	perc.	industry	chemical

'64% chemical industry'

```
  ┌──╼ **64 NUM nummod**
─┾ **proc X obj <------------------- synt. head**
  ┾──╼ **. PUNCT punct**
  └─┮ **przemysłu NOUN nmod:poss <=== annot. head**
    └──╼ **chemicznego ADJ amod**
```

(15)

3	*proc.*	*kupowanego*	*towaru*
3	perc.	purchased	goods

'3% purchased goods'

```
  ┌──╼ **3 NUM nummod <====== annot. head**
─┾ **proc X nmod:poss <----- synt. head**
  ┾──╼ **. PUNCT punct**
  │ ┌──╼ **kupowanego ADJ acl**
  └─┶ **towaru NOUN nmod:poss**
```

(16)

группа	***активистов***	*занялась*	*строительством*
group	activists.GEN	took up	construction

катапульты
catapult.GEN

'a group of activists took up the construction of the catapult'

```
  ┌─┮ **группа NOUN nsubj <-------- synt. head**
  │ └──╼ **активистов NOUN nmod <=== annot. head**
─┾ занялась VERB root
  └─┮ строительством NOUN obl
    └──╼ катапульты NOUN nmod
```

(17)

группа	***учёных***	*планировала*	*провести*	*наблюдения*
group	scientists.GEN	planned	to conduct	observations

'a group of scientists planned to conduct observations'

```
  ┌─┮ **группа NOUN nsubj <=== annot./synt. head**
  │ └──╼ **учёных NOUN nmod
─┾ занялась VERB parataxis
  └─┮ провести VERB xcomp
    └──╼ наблюдения NOUN obj
```

Quantifiers as determiners. Interestingly, we find quite a lot of cases of quantifiers in the syntactic position of determiners (*some, most, each, half* and even *no*). They are heads neither from the syntactic nor the semantic point of view. However, in some cases they are marked as heads in manual annotations, e.g. in *most analysts* in Example 18, *half the total* in Example 19, *some investors*, *each bond*, *no trading* (all from English-ARRAU) and in *несколько серых пятен /some grey spots/* from Russian-RuCor.

(18) ***Most analysts*** *had expected a sharper decline after the steep rise*

```
  ┌─╼ **Most ADJ amod <======== annot. head**
┌─┶ **analysts NOUN nsubj <---- synt. head**
├─╼ had AUX aux
╾┼ expected VERB root
│ ┌─╼ a DET det
│ ├─╼ sharper ADJ amod
├─┶ decline NOUN obj
│ ┌─╼ after ADP case
│ ├─╼ the DET det
│ ├─╼ steep ADJ amod
└─┶ rise NOUN obl
```

(19) *more than* ***half the trust's total of $268 million***

```
  ┌─┮ more ADJ advmod
  │ └─╼ than ADP fixed
┌─┶ **half PRON det:predet <==== annot. head**
│ ┌─╼ **the DET det**
├─┾ **trust NOUN nmod:poss**
│ └─╼ **'s PART case**
╾┼ **total NOUN appos <---------- synt. head**
│ ┌─╼ **of ADP case**
└─┾ **$ SYM nmod**
  │ ┌─╼ **268 NUM compound**
  └─┶ **million NUM nummod**
```

The reasons for such mismatches may be twofold. First, these constructions are not clearly distinct from the structures like 'most of people, 'half of people' where *most* and *half* are syntactic heads. Another reason may be higher salience of the determiners in the given contexts.

Dates. Mismatches in dates seem to appear only in Polish-PCC. Years and months (if present) are consistently annotated as mention heads, as illustrated in Examples 20 and 21, respectively. Therefore, it should not be too difficult to obtain such mention heads using a rule-based transformation based on syntax.

(20)

w	***1998***	***roku***	*nową*	*umowę*	*podpisała*	*żona*	*pana*	*Mariana*
in	1998	year	new	contract	sign	wife	Mr.	Marian's

'in 1998, a new contract was signed by Mr. Marian's wife '

```
    ┌─╼ w ADP case
    ├─╼ **1998 ADJ amod <====== annot. head**
  ┌─┶ **roku NOUN obl <-------- synt. head**
  │ ┌─╼ nową ADJ amod
  ├─┶ umowę NOUN obj
╾─┾ podpisała VERB root
  └─┮ żona NOUN nsubj
    └─┮ pana NOUN nmod:poss
      └─╼ Mariana PROPN flat
```

(21)

wystawa	*czynna*	*do*	***9***	***kwietnia***	***2007***
exhibition	open	until	9	April	2007

'the exhibition is open until April 9, 2007'

```
╾┮ wystawa NOUN appos
 ├─╼ czynna ADJ amod
 │ ┌─╼ do ADP case
 └─┾ **9 ADJ nmod <------------------ synt. head**
   └─┮ **kwietnia NOUN nmod:poss <=== annot. head**
     └─╼ **2007 ADJ amod**
```

6 Conclusion

We have provided a novel comparison of syntactic dependency structure on one hand, and annotation of coreferential mentions on the other hand. In particular, we focus on the notion of mention heads in coreference datasets where such a notion exists and it is not designed to be identical to the syntactic head. Nevertheless, we can compare mention heads with syntactic heads thanks to the CorefUD collection, which contains coreference corpora with dependency structures predicted by the UDPipe parser. We collected mention instances where the syntactic head did not match the designated mention head, then we manually examined a subset of such instances and analyzed the likely reasons for the difference.

If we summarize our observations, the UD heads and manually annotated mention heads coincide in majority of multi-token mentions in all four studied datasets already now, while most differences can be attributed to one of the following reasons:

- heads of a mention are different because of an error made by the UD parser, or because of an error made by an annotator; the amount of parsing errors is surprisingly low, likely due to relative simplicity of parsing of noun phrases (and will hopefully further fade out with progress in parsing technology),
- heads are selected using rather technical than linguistic rules in expressions such as named entities or coordination structures (in which linguistic intuitions for heads are weak); rule-based transformations could be used for translating UD convention to a coreference dataset convention or *vice versa*,
- semantic rather than syntactic heads are chosen in coreference annotations, e.g. in expressions with numerals; however, with an exception of some types of expressions (e.g. 'containers'), again a few rule-based patterns on the UD tree of a mention could be used to automatically identify the semantic head,

- in some cases, mention head annotations in coreference datasets bear information that seems intuitively semantically salient (such as contrast) and undeducible from UD syntax; however, such cases are rare and typically not supported by coreference annotation guidelines.

Let us conclude by answering the question from the title. It seems that both inter-project and intra-project consistency would be gained and almost nothing would be lost if we start adhering to the UD notion of heads in mentions in coreference projects, instead of annotating coreference-specific heads. In addition, quality of mention heads derived from automatic UD parses based on modern parsing technology is quite high, which would further reduce potential benefits of manual annotation of mention heads in future coreference-oriented projects.

Acknowledgements

This work was supported by the Grants GA19-14534S and 20-16819X (LUSyD) of the Czech Science Foundation; LM2018101 (LINDAT/CLARIAH-CZ) of the Ministry of Education, Youth, and Sports of the Czech Republic; and EC/H2020/825303 (Bergamot) of the European Commision.

We thank the three anonymous reviewers for their very insightful and useful comments.

References

Gosse Bouma and Gertjan van Noord. 2017. Increasing return on annotation investment: The automatic construction of a Universal Dependency treebank for Dutch. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 19–26, Gothenburg, Sweden, May. Association for Computational Linguistics.

Peter Bourgonje and Manfred Stede. 2020. The Potsdam commentary corpus 2.2: Extending annotations for shallow discourse parsing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1061–1066, Marseille, France, May. European Language Resources Association.

Marie-Catherine de Marneffe, Christopher Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Kira Droganova, Olga Lyashevskaya, and Daniel Zeman. 2018. Data Conversion and Consistency of Monolingual Corpora: Russian UD Treebanks. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)*, pages 53–66, Linköping, Sweden. Linköping University Electronic Press.

Micha Elsner and Eugene Charniak. 2010. The same-head heuristic for coreference. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 33–37.

Jan Hajič, Eduard Bejček, Jaroslava Hlaváčová, Marie Mikulová, Milan Straka, Jan Štěpánek, and Barbora Štěpánková. 2020. Prague Dependency Treebank - Consolidated 1.0. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, pages 5208–5218, Marseille, France. European Language Resources Association.

Iris Hendrickx, Gosse Bouma, Frederik Coppens, Walter Daelemans, Veronique Hoste, Geert Kloosterman, Anne-Marie Mineur, Joeri Van Der Vloet, and Jean-Luc Verschelde. 2008. A coreference corpus and resolution system for Dutch. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA).

Lynette Hirschman and Nancy Chinchor. 1998. Appendix F: MUC-7 coreference task definition (version 3.0). In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998.*

Matthias T. Kromann, Line Mikkelsen, and Stine Kern Lynge. 2003. Danish Dependency Treebank. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, pages 217–220.

Ekaterina Lapshinova-Koltunski, Christian Hardmeier, and Pauline Krielke. 2018. ParCorFull: A parallel corpus annotated with full coreference. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Igor Aleksandrovič Mel'čuk et al. 1988. *Dependency syntax: theory and practice*. SUNY press.

Anna Nedoluzhko, Michal Novák, Silvie Cinková, Marie Mikulová, and Jiří Mírovský. 2016. Coreference in Prague Czech-English Dependency Treebank. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 169–176, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský, and Daniel Zeman. 2021. Coreference meets Universal Dependencies – a pilot experiment on harmonizing coreference datasets for 11 languages. Technical Report 66, ÚFAL MFF UK, Praha, Czechia.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, pages 4034–4043, Marseille, France. European Language Resources Association.

Maciej Ogrodniczuk, Katarzyna Glowińska, Mateusz Kopeć, Agata Savary, and Magdalena Zawisławska. 2013. Polish coreference corpus. In *Human Language Technology. Challenges for Computer Science and Linguistics - 6th Language and Technology Conference, LTC 2013, Poznań, Poland, December 7-9, 2013. Revised Selected Papers*, volume 9561 of *Lecture Notes in Computer Science*, pages 215–226. Springer.

Maciej Ogrodniczuk, Katarzyna Głowińska, Mateusz Kopeć, Agata Savary, and Magdalena Zawisławska. 2015. *Coreference in Polish: Annotation, Resolution and Evaluation*. Walter De Gruyter.

Timothy Osborne, Michael Putnam, and Thomas Groß. 2012. Catenae: Introducing a novel unit of syntactic analysis. *Syntax*, 15(4):354–396.

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

Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015. A joint framework for coreference resolution and mention head detection. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 12–21.

Martin Popel, David Mareček, Jan Štěpánek, Daniel Zeman, and Zdeněk Žabokrtský. 2013. Coordination structures in dependency treebanks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 517–527, Sofija, Bulgaria. Association for Computational Linguistics.

Martin Popel, Zdeněk Žabokrtský, Anna Nedoluzhko, Michal Novák, and Daniel Zeman. 2021. Do UD Trees Match Mention Spans in Coreference Annotations? In *Findings of EMNLP 2021*. Association for Computational Linguistics.

Marta Recasens and M. Antònia Martí. 2010. AnCora-CO: Coreferentially Annotated Corpora for Spanish and Catalan. *Lang. Resour. Eval.*, 44(4):315–345, December.

Petr Sgall. 1998. Teorie valence a její formální zpracování. *Slovo a slovesnost*, 59(1):15–29.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium, October. Association for Computational Linguistics.

Svetlana Toldova, Anna Roytberg, Alina Ladygina, Maria Vasilyeva, Ilya Azerkovich, Matvei Kurzukov, G. Sim, D.V. Gorshkov, A. Ivanova, Anna Nedoluzhko, and Yulia Grishina. 2014. Evaluating Anaphora and Coreference Resolution for Russian. In *Komp'juternaja lingvistika i intellektual'nye tehnologii. Po materialam ezhegodnoj Mezhdunarodnoj konferencii Dialog*, pages 681–695.

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Francesca Delogu, Kepa J. Rodriguez, and Massimo Poesio. 2020. Annotating a broad range of anaphoric phenomena, in a variety of genres: the ARRAU Corpus. *Natural Language Engineering*, 26(1):95–128, January.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning Global Features for Coreference Resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California, June. Association for Computational Linguistics.

Amir Zeldes. 2017. The GUM Corpus: Creating Multilayer Resources in the Classroom. *Language Resources and Evaluation*, 51(3):581–612, September.

Voldemaras Žitkus and Rita Butkienė. 2018. Coreference annotation scheme and corpus for Lithuanian language. In *Fifth International Conference on Social Networks Analysis, Management and Security, SNAMS 2018, Valencia, Spain, October 15-18, 2018*, pages 243–250. IEEE.

Number agreement, dependency length, and word order in Finnish traditional dialects

Kaius Sinnemäki
University of Helsinki
P.O. Box 24
00014 University of Helsinki
kaius.sinnemaki@helsinki.fi

Akira Takaki
University of Helsinki
P.O. Box 24
00014 University of Helsinki
akira.takaki@helsinki.fi

Abstract

In this paper, we research the interaction of number agreement, dependency length, and word order between the subject and the verb in Finnish traditional dialects. While in standard Finnish the verb always agrees with the subject in person and number, in traditional dialects it does not always agree in number with a third person plural subject. We approach this variation with data from The Finnish Dialect Syntax Archive, focusing here on plural lexical subjects. We use generalized linear mixed effects modelling to model variation in number agreement and use as as a predictor the dependency length between the subject and the verb, building in word order as part of this measure. Variation across lemmas, individuals, and dialects is addressed via random grouping factors. Finite verb and the main lexical verb are considered as alternative reference points for dependency length and agreement. The results suggest that the probability of number agreement increases as the distance of the preverbal subject from the verb increases, but the trend is the opposite for postverbal subjects so that the probability of number agreement decreases as the distance of the subject from the verb increases.

1 Introduction

Over the past two decades dependency relations have been much researched from the perspective of dependency length. Dependency length measures the distance between the head and the dependent of a construction in terms of the number of intervening words. Cross-linguistic research suggests a tendency to keep dependency length minimal across languages (Hawkins, 2004; Liu et al., 2017; Gibson et al., 2019; Jing et al., to appear). Interaction of dependency length with other grammatical factors, such as word order, has also been increasingly researched. However, there has been very little research on the possible relationship between dependency length and variation in case marking and/or agreement (Ros et al., 2015; Sinnemäki and Haakana, 2021) despite increasing calls for doing so. Most previous research also focuses on written language or a mixture of spoken and written language using, for instance, the Universal Dependencies data (Zeman et al., 2021; de Marneffe et al., 2021).

In this paper we discuss the interaction of number agreement on the verb and the length of dependency between the lexical subject and the verb in Finnish traditional dialects, thus focusing on spoken language varieties. Verbs in standard spoken Finnish agree obligatorily with the subject in person and number, as in example (1a), so that using the singular form of the verb with plural subjects is ungrammatical in the standard language. However, third person plural subjects do not always trigger plural agreement on the verb in colloquial speech and in dialects, as in example (1b).

(1) a. lapse-t syö-*ø/vät
child-PL.NOM eat-3SG/3PL
'children are eating'

b. lapse-t syö-ø/vät
child-PL.NOM eat-3SG/3PL
'children are eating'

Previous work on this variation has suggested that plural agreement on the verb may be affected by different factors. These include sociolinguistic factors, such as speakers gender and dialect, as well as structural factors. For instance, plural agreement is rare with the copula verb, quite common with preverbal subjects, and quite likely when the subject is far removed from the verb (Karlsson, 1966; Karlsson, 1977; Mielikäinen, 1984). This earlier research thus already suggests that dependency length and word order affect plural agreement. There is also much cross-dialectal variation in number agreement, the plural agreement being the most frequent in the South-Eastern, the South-Western, and the Northernmost dialects but uncommon elsewhere. However, the relative effect of these factors have not been evaluated with one another using computational modelling, taking into account dialectal variation as well.

In this paper we focus on the interaction of number agreement, word order, and dependency length using corpus data on Finnish traditional dialects and modelling variation in agreement computationally. We are specifically interested in how word order and dependency length may affect variation in number agreement. While number agreement in Finnish varies in different constructions, we focus here on number agreement on the verb, because this variation is well-covered in earlier literature and provides an interesting foundation for further research.

We take as a starting point the noisy channel hypothesis, according to which language users are sensitive to how noise may corrupt the linguistic signal (Gibson et al., 2013). In the case of the dependency relation between the subject and the verb, one source of noise are words that intervene between the subject and the verb. The more such intervening words there are, the more this burdens the memory and may hamper the hearer's ability to recover the dependency relation. When applied to variable plural agreement, the noisy channel hypothesis predicts that the greater the distance between the plural subject and the verb, the more likely the verb will agree with the plural subject to maximize the hearer's ability to recover the dependency relation. But when the subject and verb are very close to each other, there is less noise from intervening words and thus the likelihood of plural agreement is predicted to be low. Other grammatical structures, such as repeating the verb, may be used for maximizing the recoverability of the dependency relation especially in spoken language, but these structures are excluded from this study.

These predictions are further qualified by word order. With plural preverbal lexical subjects, agreement is the only reliable source of information for the dependency relation in Finnish, since both plural lexical subject and objects may be in the nominative case. Because the order of subject and verb is very flexible in Finnish dialects (see Section 2), word order is not informative about syntactic structure either. However, the verb's argument structure may provide information about the arguments at the verb. Given these sources of information for recovering the dependency relation, we predict that plural agreement is more likely with preverbal than with postverbal subjects. This prediction accords also with what is known about plural agreement in the world's languages. Based on earlier research there is a universal tendency to suspend plural agreement between the subject and the verb in postverbal contexts (Greenberg, 1966), that is, to use singular verb forms with postverbal plural subjects. This pattern is found in standard Finnish as well (Karlsson, 1977).

We model the effect of dependency length on the variation in number agreement with generalized linear mixed effects modelling. The null hypothesis is that dependency length has no effect on number agreement. In the modelling we take into account variation in word order and address variation in number agreement across speakers, dialects, and lemmas as well. The data comes from roughly 4 500 clauses retrieved from The Finnish Dialect Syntax Archive (University of Turku, School of Languages and Translation Studies and Institute for the Languages of Finland, 1985). In the following, we first discuss the data and methods (Section 2), followed by the results of the statistical modelling (Section 3) and a brief discussion of the results (Section 4).

2 Data and methods

Based on earlier research variation in number agreement is particularly common in Finnish traditional dialects. For this reason, we analysed data from The Finnish Dialect Syntax Archive (University of Turku, School of Languages and Translation Studies and Institute for the Languages of Finland, 1985), which contains recorded spoken data from more than 100 interviewees, totaling roughly one million lemmas.[1] The data has been collected between the 1950s and 1970s and contains largely narratives from uneducated rural residents whose speech has not been affected by the standard language (Ikola, 1985). The interviewees' median year of birth was 1884, so the data represents Finnish dialects as learned at the end of the 19th century when standard language was taking shape but had not had a widespread effect on the population. The Archive's data is grammatically annotated and contains information, for instance, on the speakers age, gender, and dialect as well as grammatical information on each word (e.g., part of speech, inflectional categories, and syntactic function).

We extracted the data using the following criteria.[2] First, we contrasted two ways of defining the head of the construction. Dependency length is analysed as the distance between the head (the verb) and the dependent (the subject). However, when the predicate is composed of several parts, each of which can agree with the subject in number, the situation becomes more complex: how should we account for number agreement on an inflecting auxiliary that is closer to the subject compared to the main lexical verb? It is plausible to assume that placing the auxiliary close to the subject would enable earlier identification of the dependency relations (Ros et al., 2015, p. 1160-1161).

In Finnish, agreement on the predicate can be expressed on three different elements. Example in Figure (1) illustrates how not only the main lexical verb (*syödä* 'to eat') can agree with the subject in number but so can the auxiliary verb (*olla* 'to be') and the negative auxiliary verb (*ei*). Such complex predicates pose a potential problem for analysing the relationship between agreement and dependency length. In this paper we contrast two ways of approaching this issue. We start by modelling the finite verb as the head, that is, as the reference point for dependency length and agreement. In the case of simple verbs the main lexical verb is also the finite verb. In the case of complex verbs, the finite auxiliary is the finite element, while the main lexical verb is non-finite. We then contrasted this approach by modelling the main lexical verb as the head. However, in the case of complex verbs with three elements, the non-finite auxiliary verb (*olla* in the example in Figure 1) could be considered as an alternative reference point for dependency length and agreement as well. This was not attempted here, since there were only 14 such instances and in each of them the auxiliary was in the singular.

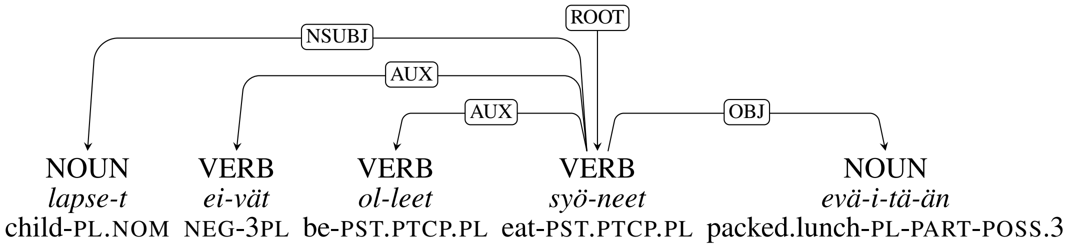


Figure 1: Dependency tree of the Finnish sentence 'The children had not eaten their packed lunches'.

Second, we limited the analysis to clauses containing a lexical subject and excluded pronoun subjects from the study. The reason for this was that earlier research on third person plural subject pronouns has already suggested that a growing distance between the subject pronoun and the main lexical verb increases the probability of plural forms on the verb at least with preverbal subjects (Sinnemäki and Haakana, 2021). There are also two third person plural pronouns in Finnish, namely *ne* and *he*. The latter pronoun is much less common across Finnish dialects but it also occurs much more frequently with plural agreement compared to *ne*. For these reasons, we thought it would be meaningful to focus only on lexical subjects and to contrast also the preverbal and the postverbal domains.

Third, while the corpus is carefully annotated for grammatical information, it does not currently code dependency relations as treebanks do. For this reason, we automatically extracted all relevant clauses and then manually double-checked each verb-subject pair for dependency length, word order, and overall

[1] The whole corpus is openly available via the Language Bank of Finland at `http://urn.fi/urn:nbn:fi:lb-2019092002`.

[2] The analysed data and the scripts are available at `https://version.helsinki.fi/gramadapt/depling2021-number-agreement`.

correctness of the analysis. In general, the greater the initial dependency length was, the more likely it was wrongly analysed by our automatic extraction. There were also some cases where the verb was repeated multiple times before the subject, which led to suspiciously long dependency lengths in the automatic analysis. In the manual analysis, the dependency length for such sentences was analysed from the nearest verb to the subject. One of the most extreme cases is illustrated in (2).

(2) *nii sitte oli tuola täälä ojala-sa justihin siittä kajuuti-lta ojalankylä sielä oli ni*
so then be.PST there here Ojala-INE right there.from Kajuuti-ABL Ojala.village there be.PST yes
oli kinkerit
be.PST reading.exams

‘So, there were reading examinations at Ojala-village, right at Kajuutti.’

In this example, the distance between the first copula *oli* and its subject *kinkerit* is 12.[3] However, the copula is repeated twice before the lexical subject and the closest copula is actually adjacent to the subject. By and large the automatic analysis of dependency lengths were correct in subject-verb orders, but in verb-subject clauses about a quarter were discarded, because the verb was preceded by another subject, often an anaphoric pronoun. This was expected to some extent, as Finnish is an SVO language. Following these criteria the final data contains 4 561 clauses.

Although the annotation of the original corpus has been meticulously refined over the years, it may still contain errors. For instance, we corrected 46 lemmas (roughly 1%) that were wrongly analysed in the original. It is possible that some subject-verb dependencies were overlooked by our automatic extraction, potentially leading to some false negatives (that is, excluding instances that should have been included). However, since our extraction method relied on the annotations, the potential false positives would most likely stem from problems in the original annotation. We did not estimate the correctness of the original annotations in this regard but suspect the rate of unrecognised dependencies is very low.

Length of dependency is defined as the number of intervening words between the head and the dependent in a construction. For the purpose of modelling, we coded dependency length following Gildea and Temperley (2010) so that it received negative values in left-branching dependency-relations, that is, where the subject preceded the verb (the finite auxiliary or the main lexical verb), and positive values in right-branching dependency-relations, that is, where the subject followed the verb, the head (the verb) itself at zero. This coding enables us to keep the ensuing model structure simple and to put emphasis on dependency length in the modelling, while still being able to inspect linear order at least visually. An alternative would have been to use positive counts for dependency length and to model its interaction with word order. Because this would have increased the complexity of the model we opted for coding dependency length with both positive and negative values.

Figure 2 displays the histogram for dependency length over agreement. In both plots, the majority of instances is adjacent to the verb with diminishing number of instances as the distance from the verb grows. The subject tends to occur mostly preverbally, but postverbal lexical subjects are also common with both finite verbs (plot A) and main lexical verbs (plot B). Overall, there is a lot of variation in the order of the subject and the verb in Finnish dialects. The distribution of number agreement is biased so that plural forms of the verb are relatively more common among preverbal lexical subjects, while singular forms are relatively more common among postverbal lexical subjects. In addition, plural agreement seems slightly more common as the preverbal subject is further removed from the verb and singular agreement seems slightly more common as the postverbal subject is further removed from the verb. Yet based on the histograms alone it is hard to draw conclusions on how number agreement behaves more generally as distance from the verb increases.

To estimate whether dependency length has an effect on number agreement, we used mixed effects logistic regression. Number agreement was modelled as a binomial response variable with values ”singular” (reference level) and ”plural”. Dependency length between the subject and the verb was modelled as a predictor, counted as the number of intervening words as stated above. Two different models were contrasted. In the first model, called here m.fin, the finite auxiliary was analysed as the head. In these

[3] *Kinkerit* refers to examinations in rural areas held historically to teach and test reading skills and knowledge of Christianity.

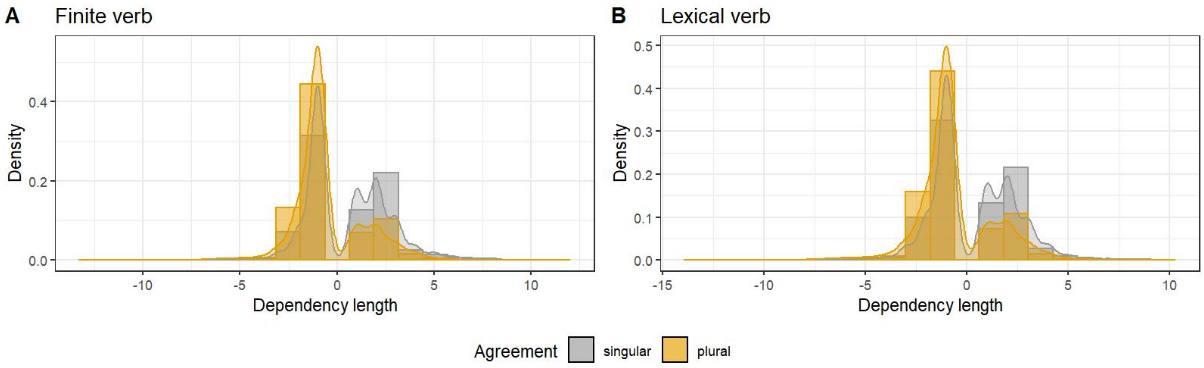


Figure 2: Histograms for dependency length over number agreement (finite verbs as the reference point in plot A and main lexical verbs in plot B).

models we analysed the occurrence of number agreement on the finite auxiliary and used it also as a reference point for counting dependency length. In the m.fin models there were 934 (21%) clauses with plural agreement; dependency length ranged from values -13 to +11, the verb being at zero. In the second model, called here m.lex, the main lexical verb was analysed as the head. In these models we analysed the occurrence of number agreement on the main lexical verb and used it also as a reference point for counting dependency length. In the m.lex models there were 1003 (23%) clauses with plural agreement; dependency length ranged from values -13 to +10, the verb being at zero.

Three random intercepts were included in both models: i. the lemma of the (main lexical) verb, ii. the lemma of the lexical subject, and iii. the individual speaker nested in their local dialect group. Based on earlier research the lemma of the verb may affect number agreement in Finnish dialects: plural agreement is particularly rare with the copula *olla*, but there is great variation across different verbs. In m.fin models there can be only two alternative finite elements, namely, the negative auxiliary *ei* or the verb *olla* which functions as an auxiliary in the perfect and pluperfect tenses. For this reason we modelled the main lexical verb as a random intercept also in the m.fin model. We also assume that variation depending on the subject lemma needs to be accounted in the modelling, analogously to the verb lemma. The hierarchic structure of embedding each speaker in their dialect group enables taking into account variation in number agreement within and across dialects and speakers.

The models were fitted in R using the package `blme` (Chung et al., 2013), which enables maximum penalized likelihood with weakly informative priors and posterior modes for estimation. It often leads to better convergence compared to `lme4` as well as drawing correlation terms away from perfect correlation. The model specification in the `lme4` notation (Bates et al., 2015) was as in (3). The p-values were drawn with likelihood ratio. The models' explanatory power was computed separately for the whole model (conditional R^2) and just for the fixed effects (marginal R^2) via the package `MuMIn` (Barton, 2020). The algorithm is based on Nakagawa and Schielzeth (2013) and has been further developed by Johnson (2014), and Nakagawa et al. (2017).[4]

(3) agreement $\sim$ dep.length + (1|lemma.noun) + (1|lemma.verb) + (1|dialect/individual)

3 Results

According to the results, dependency length had a significant negative effect on plural agreement when finite verbs were selected as the reference point ($estimate = -0.28 \pm 0.03; \chi^2(1) = 97.2; p < 0.001$). This means that as dependency length increases by one unit, the likelihood of plural agreement on the finite verb decreases about 1.25 times. When selecting the main lexical verb as the reference point, dependency length had also a significant negative effect on plural agreement ($estimate = -0.18 \pm$

[4] The R package `tidyverse` (Wickham et al., 2019) was used in preprocessing the data in R; graphics were computed using packages `sjPlot` (Lüdecke, 2020), `cowplot` (Wilke, 2020), and `ggplot2` (Wickham, 2016).

$0.03; \chi^2(1) = 39.7; p < 0.001$). This means that as dependency length increases by one unit, the likelihood of plural agreement on the main lexical verb decreases about 1.17 times.

We evaluated the models' goodness-of-fit with Akaike Information Criterion (AIC) by comparing the difference in the nested models' values for AIC. Adding dependency length to the null model m.fin lowers AIC by 95, while adding dependency length to the null model m.lex lowers AIC by 38. This large reductions in AIC (> 10) provide evidence for both models' goodness (Burnham and Anderson, 2002, p. 70-71). The explanatory power of dependency length in model m.fin was about 0.030 (marginal R^2) and for the whole model about 0.494 (conditional R^2); for model m.lex the respective figures were 0.013 and 0.500. Accordingly, most of the variation in plural agreement was explained by dialectal and individual differences, but even so the models were able to recognize a small effect for dependency length.

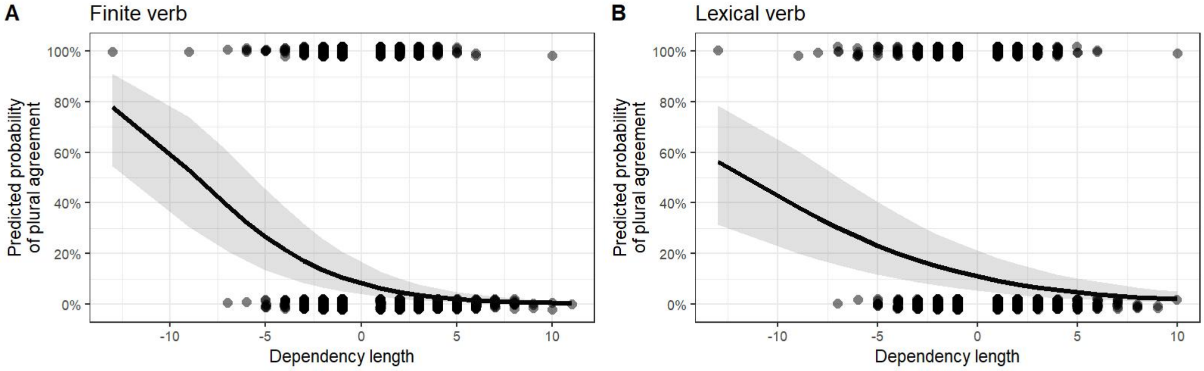


Figure 3: Marginal effects for dependency length over number agreement (in plot A for finite verbs and in plot B for main lexical verbs; small jitter is added to the datapoints).

Figure 3 presents the marginal effect plots for the two models. The plots suggest a clear inverse relationship between dependency length and number agreement. In both plots the predicted probability of plural agreement is about 10% when the plural lexical subject is adjacent to the verb. However, the more words intervene between a *preverbal* subject and the verb, the greater the predicted probability of plural agreement becomes. In plot A it is around 40% at a distance of seven and increases above 60% at the greatest distances, while in plot B it is around 30% at a distance of seven and increases above 40% at the greatest distances. On the other hand, the more words intervene between a *postverbal* lexical subject and the verb, the smaller and ever closer to zero the predicted probability of plural agreement becomes in both plots. Word order thus seems to condition the effect of dependency length on number agreement: plural agreement is more likely when the lexical subject precedes the verb than when it follows the verb, and the difference between the word orders becomes the clearer the greater the dependency length is.

4 Discussion

Based on our analyses, there was an inverse relationship between number agreement and dependency length in Finnish traditional dialects partly conditioned by word order. The inverse relationship was a little stronger with finite verbs than with main lexical verbs. But regardless of which was taken as the reference point for agreement and dependency length, the results were significant and very similar.

Since our models were random intercept models we could not estimate whether dependency length had a similar effect on agreement across dialects. To evaluate this, we fitted two further models. These models were otherwise identical to the random intercept models, but we fitted a random slope for dependency length over dialect groups (and over individuals). Because plural agreement is very unevenly distributed across dialects, we included data from only those dialect groups in which there were 20 or more instances of plural agreement and where that incidence was 10% or more of all the instances.

According to the results, dependency length had a significant negative effect on plural agreement with finite verbs ($estimate = -0.37 \pm 0.08; \chi^2(1) = 13.0; p < 0.001$) as well as with main lexical verbs ($estimate = -0.32 \pm 0.09; \chi^2(1) = 10.4; p = 0.0013$). The marginal effects in Figure 4 are quite similar across the dialects regardless of using finite verbs (plot A) or main lexical verbs (plot B) as reference points for dependency length and number agreement: the farther a *preverbal* lexical subject is removed from the verb, the *more likely* there is plural agreement on the verb, and the farther a *postverbal* lexical subject is removed from the verb, the *less likely* there is plural agreement on the verb. These results suggest the relationship between agreement and dependency length is similar across the traditional Finnish dialects and regardless of which verb was selected as the reference point.

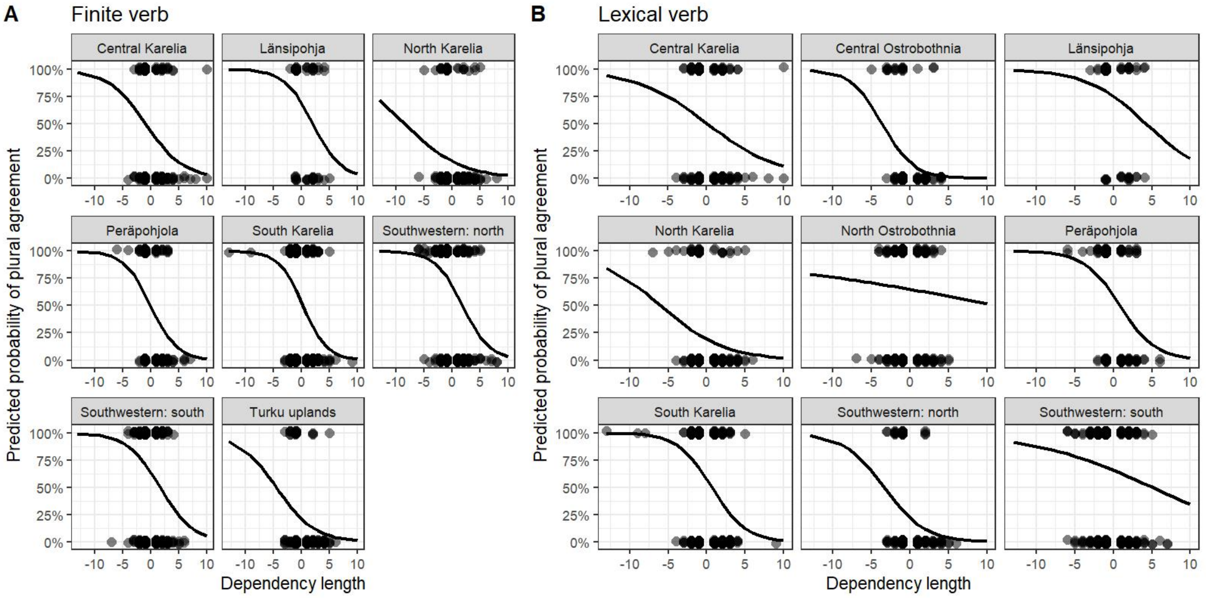


Figure 4: Marginal effects for dependency length over number agreement in the random slope models.

The results largely support our predictions based on the noisy channel hypothesis. Plural agreement increased in probability as more words intervened between the subject and the verb. This result aligns with earlier research on third person plural pronoun subjects in Finnish (Sinnemäki and Haakana, 2021). We also predicted that plural agreement would be less likely with postverbal subjects compared to preverbal subjects, and the results provide evidence for this hypothesis as well.

However, it was somewhat unexpected that the probability of plural agreement became increasingly smaller the farther the postverbal lexical subject was removed from the verb. While the results align with how other languages work (Greenberg, 1966), it is unclear why plural agreement would be less likely with postverbal subjects far removed from the verb compared to postverbal subjects that were adjacent to the verb. In the postverbal contexts in Finnish, the subject may be more easily confused with the object, because direct objects tend to occur postverbally and since plural lexical objects as well as plural lexical subjects may occur in the nominative case (objects also in the partitive case). It would thus seem that there were more possibilities for confusing the subject and the object in the postverbal domain, which, according to the noisy channel hypothesis, would call for increased probability of agreement with postverbal subjects, at least for transitive and ditransitive verbs. Further research is needed to determine which factors affect variation in plural agreement especially in the postverbal domain.

The results raise a more general question whether the observed relationship between number agreement and dependency length is limited to Finnish dialects or a more general tendency in languages. We do not consider it implausible that number agreement and dependency length would pattern in similar ways in other languages as well, but this remains as an issue for future research, since the interaction between dependency length and agreement has not yet been widely researched across languages.

Acknowledgements

This research has received funding by the European Research Council (ERC), grant no 805371 to Kaius Sinnemäki (PI). We are grateful to three anonymous reviewers and to the editors for comments.

References

Kamil Barton. 2020. Mumin: Multi-model inference. r package version 1.43.17.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Kenneth P. Burnham and David R. Anderson. 2002. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer, New York, second edition.

Yeojin Chung, Sophia Rabe-Hesketh, Vincent Dorie, Andrew Gelman, and Jingchen Liu. 2013. A nondegenerate penalized likelihood estimator for variance parameters in multilevel models. *Psychometrika*, 78(4):685–709.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational Linguistics*, 47(2):255–308.

Edward Gibson, Steven T. Piantadosi, Kimberly Brink, Leon Bergen, Eunice Lim, and Rebecca Saxe. 2013. A noisy-channel account of crosslinguistic word-order variation. *Psychological Science*, 24(7):1079–1088.

Edward Gibson, Richard Futrell, Steven P. Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. 2019. How efficiency shapes human language. *Trends in Cognitive Sciences*, 23(5):389–407.

Daniel Gildea and David Temperley. 2010. Do grammars minimize dependency length? *Cognitive Science*, 34(2):286–310.

Joseph H. Greenberg. 1966. Some universals of grammar with particular reference to the order of meaningful elements. In Joseph H. Greenberg, editor, *Universals of Language*, pages 73–113. MIT Press, Cambridge, MA, second edition.

John A. Hawkins. 2004. *Efficiency and Complexity in Grammars*. Oxford University Press, Oxford.

Osmo Ikola, editor. 1985. *Lauseopin arkiston opas*, volume 1 of *Lauseopin arkiston julkaisuja*. Turun yliopisto, Turku.

Yingqi Jing, Damin E. Blasi, and Balthasar Bickel. to appear. Dependency length minimization and its limits: a possible role for a probabilistic version of the final-over-final condition. *Language*.

Paul C.D. Johnson. 2014. Extension of nakagawa & schielzeth's r2glmm to random slopes models. *Methods in Ecology and Evolution*, 5(9):944–946.

Göran Karlsson. 1966. Eräitä tilastollisia tietoja subjektin ja predikaatin numeruskongruenssista suomen murteissa. *Sananjalka*, 8:2–23.

Fred Karlsson. 1977. Syntaktisten kongruenssijärjestelmien luonteesta ja funktioista. *Virittäjä*, 81(4):359–391.

Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of Life Reviews*, 21:171–193.

Daniel Lüdecke, 2020. *sjPlot: Data Visualization for Statistics in Social Science*. R package version 2.8.4.

Aila Mielikäinen. 1984. Monikon 3. persoonan kongruenssista puhekielessä. *Virittäjä*, 88(2):162–175.

Shinichi Nakagawa and Holger Schielzeth. 2013. A general and simple method for obtaining R^2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2):133–142.

Shinichi Nakagawa, Paul C.D. Johnson, and Holger Schielzeth. 2017. The coefficient of determination R^2 and intra-class correlation coefficient from generalized linear mixed-effects models revisited and expanded. *Journal of the Royal Society Interface*, 14(134):20170213.

Idoia Ros, Mikel Santesteban, Kumiko Fukumora, and Itziar Laka. 2015. Aiming at shorter dependencies: The role of agreement morphology. *Language, Cognition and Neuroscience*, 30(9):1156–1174.

Kaius Sinnemäki and Viljami Haakana. 2021. Variationistinen korpustutkimus predikaatin differentiaalisesta lukukongruenssista ja substantiiviluokasta suomen murteissa. In Leena Maria Heikkola, Geda Paulsen, Katarzyna Wojciechowicz, and Jutta Rosenberg, editors, *Språkets funktion: Juhlakirja Urpo Nikanteen 60-vuotispäivän kunniaksi-Festskrift till Urpo Nikanne på 60-årsdagen-Festschrift for Urpo Nikanne in honor of his 60th birthday*, pages 96–130. Åbo Akademis förlag, Åbo.

University of Turku, School of Languages and Translation Studies and Institute for the Languages of Finland. 1985. The Finnish Dialect Syntax Archive's Helsinki Korp Version.

Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain Franois, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Mller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. 2019. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686.

Hadley Wickham. 2016. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York.

Claus O. Wilke, 2020. *cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'*. R package version 1.1.1.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, et al. 2021. Universal dependencies 2.8.1. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.