TLT 2025

23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025)

Proceedings

August 28-29, 2025

The TLT organizers gratefully acknowledge the support from the following sponsors.

VITASIS

USTANOVA PATRA STANISLAVA ŠKRABCA

UniDive

Ljubljana
Ljubljana Tourism

Mestna občina Ljubljana

Flanders
State of the Art

CLARIN.SI

cjvt Centre for Language Resources and Technologies

AMEBIS

alpineon

AI4DH CENTRE OF EXCELLENCE IN AI FOR DIGITAL HUMANITIES

Organized by

FRI
UNIVERSITY OF LJUBLJANA
Faculty of Computer and Information Science

FF
UNIVERSITY OF LJUBLJANA
Faculty of Arts

SDJT
Slovensko društvo za jezikovne tehnologije

As part of SyntaxFest 2025

SYNTAXFEST

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-291-6

Introduction

The 23rd International Workshop on Treebanks and Linguistics (TLT) follows an annual series that started in 2002, in Sozopol, Bulgaria. TLT addresses all aspects of treebank design, development, and use, "Treebank" is taken in a broad sense, comprising any spoken, signed, or written data augmented with computationally processable annotations of linguistic structure at various levels. This year, TLT took place at SyntaxFest 2025 in Ljubljana, Slovenia, which brought together five related but independent events:

- 18th International Conference on Parsing Technologies (IWPT 2025)
- 8th Universal Dependencies Workshop (UDW 2025)
- 8th International Conference on Dependency Linguistics (DepLing 2025)
- 23rd Workshop on Treebanks and Linguistic Theories (TLT 2025)
- 3rd Workshop on Quantitative Syntax (QUASY 2025)

In addition, a pre-conference workshop organized by the COST Action CA21167 – Universality, Diversity and Idiosyncrasy in Language Technology (UniDive) was held prior to the main event, with dedicated sessions on the 1st UniDive Shared Task on Morphosyntactic Parsing and the 2nd Workshop on Universal Dependencies for Turkic Languages.

SyntaxFest 2025 continues the tradition of SyntaxFest 2019 (Paris, France), SyntaxFest 2021 (Sofia, Bulgaria), and GURT/SyntaxFest 2023 (Washington DC, USA) in bringing together multiple events that share a common interest in using corpora and treebanks for empirically validating syntactic theories, studying syntax from quantitative and theoretical points of view, and training machine learning models for natural language processing. Much of this research is increasingly multilingual and cross-lingual and requires continued systematic analysis from various theoretical, applied, and practical perspectives. By co-locating these workshops under a shared umbrella, SyntaxFest fosters dialogue between overlapping research communities and supports innovation at the intersection of linguistics and language technology.

As in previous editions, all five workshops at SyntaxFest 2025 shared a common submission and reviewing process, with a unified timeline, identical submission formats, and a shared program committee. During submission, authors could indicate one or more preferred venues, but the final assignment of papers was determined by the collective program chairs, composed of the individual workshop chairs, based on thematic alignment. All accepted submissions were peer-reviewed by at least three reviewers from the shared program committee.

In total, SyntaxFest 2025 received 94 submissions, of which 73 (78%) were accepted for presentation. The final program included a total of 47 long papers, 21 short papers, and 5 non-archival contributions, distributed across the five workshops: 5 papers were presented at IWPT (2 long, 3 short); 20 at UDW (14 long, 5 short, 1 non-archival); 16 at DepLing (12 long, 2 short, 2 non-archival); 18 at TLT (10 long, 7 short, 1 non-archival); and 14 at QUASY (9 long, 4 short, 1 non-archival).

Our sincere thanks go to everyone who made this event possible. We thank all authors for their submissions and the reviewers for their time and thoughtful feedback, which contributed to a diverse and high-quality program. Special thanks go to the local organizing team at the University of Ljubljana and the Slovene Language Technologies Society for hosting the event, and to the sponsors for their generous support. Finally, we gratefully acknowledge ACL SIGPARSE for endorsing the event and the ACL Anthology for publishing the proceedings.

Kenji Sagae, Stephan Oepen (IWPT 2025 Chairs)
Gosse Bomma, Çağrı Çöltekin (UDW 2025 Chairs)
Eva Hajičová, Sylvain Kahane (DepLing 2025 Chairs)
Heike Zinsmeister, Sarah Jablotschkin, Sandra Kübler (TLT 2025 Chairs)

Xinying Chen, Yaqin Wang (QUASY 2025 Chairs)
Kaja Dobrovoljc (SyntaxFest 2025 Organization Chair)

Ljubljana, August 2025

Organizing Committee

TLT Chairs

Heike Zinsmeister, University of Hamburg
Sarah Jablotschkin, University of Hamburg
Sandra Kübler, Indiana University

DepLing Chairs

Eva Hajičová, Charles University, Prague
Sylvain Kahane, Université Paris Nanterre

UDW Chairs

Gosse Bomma, University of Groningen
Çağrı Çöltekin, University of Tübingen

IWPT Chairs

Kenji Sagae, University of California, Davis
Stephan Oepen, University of Oslo

QUASY Chairs

Xinying Chen, University of Ostrava
Yaqin Wang, Guangdong University of Foreign Studies

Publication Chair

Sarah Jablotschkin, University of Hamburg

Local SyntaxFest 2025 Organizing Committee

Kaja Dobrovoljc, University of Ljubljana, SDJT
Špela Arhar Holdt, University of Ljubljana
Luka Terčon, University of Ljubljana
Marko Robnik-Šikonja, University of Ljubljana
Matej Klemen, University of Ljubljana
Sara Kos, University of Ljubljana
Timotej Knez, University of Ljubljana, SDJT
Tinca Lukan, University of Ljubljana

Special Thanks for designing the SyntaxFest 2025 logo to

Kim Gerdes, Université Paris-Saclay

Program Committee

Shared Program Committee

V.S.D.S.Mahesh Akavarapu, Eberhard-Karls-Universität Tübingen
Leonel Figueiredo de Alencar, Federal University of Ceará (UFC)
Patricia Amaral, Indiana University
Giuseppe Attardi, University of Pisa
John Bauer, Stanford University
David Beck, University of Alberta
Laura Becker, Albert-Ludwigs-Universität Freiburg
Aleksandrs Berdicevskis, Gothenburg University
Ann Bies, University of Pennsylvania
Igor Boguslavsky, Universidad Politécnica de Madrid
Bernd Bohnet, Google
Cristina Bosco, University of Turin
Gosse Bouma, University of Groningen
Miriam Butt, Universität Konstanz
G. A. Celano, Universität Leipzig
Heng Chen, Guangdong University of Foreign Studies
Xinying Chen, University of Ostrava
Jinho D. Choi, Emory University
Çağrı Çöltekin, University of Tuebingen
Daniel Dakota, Leidos
Stefania Degaetano-Ortlieb, Universität des Saarlandes
Kaja Dobrovoljc, University of Ljubljana
Jakub Dotlacil, Utrecht University
Gülşen Eryiğit, Istanbul Technical University
Kilian Evang, Heinrich Heine University Düsseldorf
Pegah Faghiri, CNRS
Ramon Ferrer-i-Cancho, Universidad Politécnica de Cataluna
Marcos Garcia, Universidade de Santiago de Compostela
Kim Gerdes, Université Paris-Saclay
Loïc Grobol, Université Paris Nanterre
Bruno Guillaume, INRIA
Carlos Gómez-Rodríguez, Universidade da Coruña
Eva Hajicova, Charles University
Dag Trygve Truslew Haug, University of Oslo
Santiago Herrera, University of Paris Nanterre
Richard Hudson, University College London
Maarten Janssen, Charles University Prague
Jingyang Jiang, Zhejiang University
Mayank Jobanputra, Universität des Saarlandes
Sylvain Kahane, Université Paris Nanterre
Václava Kettnerová, Charles University Prague
Sandra Kübler, Indiana University
Guy Lapalme, University of Montreal
François Lareau, Université de Montréal
Miryam de Lhoneux, KU Leuven
Zoey Liu, University of Florida

Teresa Lynn, Dublin City University
Jan Macutek, Slovak Academy of Sciences
Robert Malouf, San Diego State University
Marie-Catherine de Marneffe, UCLouvain
Nicolas Mazziotta, Université de Liège
Alexander Mehler, Johann Wolfgang Goethe Universität Frankfurt am Main
Maitrey Mehta, University of Utah
Wolfgang Menzel, Universität Hamburg
Marie Mikulová, Charles University
Aleksandra Miletić, University of Helsinki
Jasmina Milićević, Dalhousie University
Simon Mille, Dublin City University
Yusuke Miyao, The University of Tokyo
Noor Abo Mokh, Indiana University
Simonetta Montemagni, Institute for Computational Linguistics "A. Zampolli" (ILC-CNR)
Jiří Mírovský, Charles University Prague
Kaili Müürisep, Institute of computer science, University of Tartu
Anna Nedoluzhko, Charles University Prague
Ruochen Niu, Beijing Language and Culture University
Joakim Nivre, Uppsala University
Stephan Oepen, University of Oslo
Timothy John Osborne, Zhejiang University
Petya Osenova, Sofia University "St. Kliment Ohridski"
Agnieszka Patejuk, Polish Academy of Sciences
Lucie Poláková, Charles University Prague
Prokopis Prokopidis, Athena Research Center
Mathilde Regnault, Universität Stuttgart
Kateřina Rysová, University of South Bohemia
Magdaléna Rysová, Charles University Prague
Tanja Samardzic, University of Zurich
Giuseppe Samo, Beijing Language and Culture University
Haruko Sanada, Rissho University
Nathan Schneider, Georgetown University
Djamé Seddah, Sorbonne University
Anastasia Shimorina, Orange
Maria Simi, University of Pisa
Achim Stein, University of Stuttgart
Daniel G. Swanson, Indiana University
Luka Terčon, Faculty of Arts, University of Ljubljana
Giulia Venturi, Institute for Computational Linguistics "A. Zampolli" (ILC-CNR)
Veronika Vincze, University of Szeged
Yaqin Wang, Guangdong University of Foreign Studies
Pan Xiaxing, Huaqiao University
Chunshan Xu, Anhui Jianzhu University
Nianwen Xue, Brandeis University
Jianwei Yan, Zhejiang University
Zdenek Zabokrtsky, Faculty of Mathematics and Physics, Charles University Prague
Eva Zehentner, University of Zurich
Amir Zeldes, Georgetown University
Daniel Zeman, Charles University Prague
Šárka Zikánová, Charles University Prague

Heike Zinsmeister, Universität Hamburg

Keynote

Subject prominence revisited: What makes entities salient?

Amir Zeldes
Georgetown University

Abstract: In this talk, I'll explore what makes certain entities stand out in discourse — what we might call more or less "salient" — and how speakers systematically identify them. Building on existing approaches to information structural "aboutness", subjecthood, Centering Theory and animacy hierarchies, I argue that salience goes beyond surface categories such as definiteness, pronominalization and grammatical function. It's also shaped by deeper structures: distributional cues, discourse relations, hierarchical organization, genre conventions, and the communicative goals we infer from context. To get at this, I use a graded notion of salience based on how often entities are included in multiple human-written summaries of a text or conversation. Drawing on manually treebanked data from 24 different spoken and written genres in English, I ask: how is salience expressed for each entity mentioned in a discourse? I'll show that while traditional linguistic markers of salience all correlate with our salience scores to some extent, every rule has exceptions, and no single feature tells the whole story. Instead, salience cuts across all levels of linguistic structure, and the most informative theoretical model of the phenomenon must therefore combine cues from across morphosyntax, discourse structure, and functional pragmatics.

Bio: Amir Zeldes is Associate Professor of Computational Linguistics at Georgetown University, where he runs the Georgetown University Corpus Linguistics lab, Corpling@GU. He has worked on multilayer treebank construction and evaluation, including development of the Georgetown University Multilayer corpus (GUM) and datasets for low resource languages, such as the UD Coptic Treebank. His main area of research is computational discourse modeling, working on frameworks such as Enhanced Rhetorical Structure Theory (eRST) and Graded Salience, as well as topics such as coreference resolution, genre variation and summarization. He is currently president of the ACL Special Interest Group on Annotation (SIGANN).

Non-Archival Abstract

Segmentation of Sino-origin words to enhance the representation of Korean and Japanese in S/UD-format treebanks

Raoul Blin[1] and Jinnam Choi[2]
[1]CNRS-CRLAO
[2]CLLE, Université Jean-Jaurès

In the Japanese and Korean S/UD treebanks, Chinese-origin words composed of two morphophonological units are not segmented, even when they are semantically transparent. We propose segmenting and annotating these words with dependency relations in order to achieve a more fine-grained and unified description of both languages. As an example, we apply this analysis to the pre-annotated GSD corpora in SUD format, and we examine the benefits and limitations of a rule-based approach.

Table of Contents

Annotation of Chinese Light Verb Constructions within UMR

Jingyi Li[1,2], Jin Zhao[2], Nianwen Xue[2], Shili Ge[1]
[1]Guangdong University of Foreign Studies
[2]Brandeis University
Corresponding author: lijingyi@mail.gdufs.edu.cn

Abstract

This paper discusses the challenges of annotating predicate-argument structures in Chinese light verb constructions (LVCs) within the Uniform Meaning Representation (UMR) framework, a cross-linguistic extension of Abstract Meaning Representation (AMR). A central challenge lies in reliably identifying LVCs in Chinese and determining their appropriate representation in UMR. We analyze the linguistic properties of Chinese LVCs, outline annotation difficulties for these structures and related constructions, and illustrate these issues through concrete examples. Our analysis focuses specifically on LVC.full types, where the light verb serves solely to convey morphological features and aspectual information. We exclude LVC.cause types, in which the light verb introduces an additional argument (e.g., a causal agent or source) to the event or state denoted by the nominal predicate. To address the practical challenge of semantic role assignment in Chinese LVCs, we propose a dual-path annotation approach: due to the compositional nature of these constructions, we recommend independently annotating the argument structure of the nominal predicate while systematically encoding the grammatical attributes and relations introduced by the light verb.

1 Introduction

The presentation of Light Verb Constructions (LVCs) continues to be a focal issue in both traditional linguistics and computational linguistics, garnering substantial attention over the years (Sag et al., 2002; Stevenson et al., 2004; Tu & Roth, 2011; Vincze et al., 2011; Nagy et al., 2020). LVCs are widely acknowledged as a universal linguistic phenomenon, composed of a verb—often referred to as "light"—paired with a single or compound predicative noun in its direct object position. The light verb makes only a minimal semantic contribution to the construction; instead, it primarily carries essential morphosyntactic properties such as person, number, tense, mood, and aspect (Savary et al., 2017; Bonn et al., 2023). Light verbs often exhibit unique and sometimes unpredictable behaviors across languages. Chinese light verbs, in particular, with their syntactic and semantic flexibility, combined with a distinctive distribution that sets them apart from regular verbs—which typically exhibit higher semantic content and more specific argument requirements—pose challenges for their identification and representation within various meaning representation frameworks. (Butt, 2010; Lin et al., 2014; Huang et al., 2014; Jiang et al., 2018; Bonn et al., 2023).

Uniform Meaning Representation (UMR), a recent graph-based framework designed to capture meaning across entire documents, provides promising opportunities for annotating LVCs, including those in Chinese, where compounding is a common word formation process (Bonn et al., 2023; Sun et al., 2023). Rooted in Abstract Meaning Representation (AMR), the fundamental components of a UMR graph are concepts and relations. At the sentence level, concepts typically map to words within a sentence, while at the document level, relations depict the semantic connections between these words (Bonn et al., 2024). At sentence level, UMR is flexible in allowing the use of both generic semantic roles, such as agent, theme, and patient, as well as predicate-specific roles, a practice widely adopted in the proposition bank approach to semantic role labeling (Xue and Palmer, 2009). Predicate-specific roles are defined in the PropBank Framesets, which provide entries for each predicate in a language (Xue and Palmer, 2005). Each predicate sense is assigned a unique set of core roles, labeled with numerical IDs prefixed by "Arg." For instance, the Chinese verb 认可 [renke, "accept"] has a first semantic frame, "认可-01," which defines the following roles:

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 1–9
August 28-29, 2025 ©2025 Association for Computational Linguistics

Arg0-agent: the entity described
Arg1-tar: the entity Arg0 accepts/ratifies

The Chinese verb 认可 [renke, “accept”] serves as an example where its defined roles can be applied to annotate occurrences of 认可 [renke, “accept”], even in contexts where some of its arguments are not explicitly stated. The LVC 获得认可 [huode-renke, “get-accept”] in (1) can be annotated in UMR as follows:

(1) 这 一 方法 获得-认可 。
this one method get-accept .
“This method got accepted.”

```
(s1x / 认可-01
    :Arg1 (s1x2 / 方法 [fangfa, “method”]
        :mod (s1x3 / 这 [zhe, “this”]))
    :Aspect Performance
    :MODSTR FullAff
```

The absence of clear morphological distinctions between certain Chinese nouns and their verbal counterparts, such as 认可 [renke, “accept”], allows these lexical items to serve both nominal and verbal roles. In example (1), we identify 认可 as the main predicate rather than 获得 [huode, “get”], annotating 方法 [fangfa, “method”] as the Arg1 of 认可 [renke, “accept”]. This annotation choice reflects that the verb 获得 [huode, “get”] acts solely as a grammatical marker indicating successful completion of the event, without adding significant semantic content.

Such light verbs have received less scholarly attention as they mostly function as regular verbs and require clear linguistic features for accurate identification. Although comparative studies have examined LVCs in English and Chinese and explored variations across major Chinese-speaking regions, such as Hong Kong, Taiwan, and Beijing (Lin et al., 2014; Huang et al., 2014; Jiang et al., 2018; Tsou and Yip, 2020; Lu, 2016), research has primarily concentrated on specific verb groups, notably “do” (做 [zuo], 干 [gan], 搞 [gao]) and “give” (加以 [jiayi], 予以 [yuyi]). Therefore, further investigation into other less commonly studied LVC types is needed to enrich both linguistic analysis and computational modeling.

The absence of clear morphological distinctions between certain Chinese nouns and their verbal counterparts, along with the intricate modifiers and argument structures of nominal complements, makes annotating Chinese LVCs particularly challenging (Wang et al., 2023). These complexities highlight the need to strike a balance that ensures consistency across different types of Chinese LVCs—a task that is both essential and demanding. We adopt broad criteria for LVC annotation from the PARSEME guidelines, a European project aimed at processing multiword expressions, including LVCs (Savary et al., 2017). Jiang et al. (2018) applied these guidelines to the automatic tagging of Chinese light verbs and introduced valuable adaptations. Nevertheless, their research mainly focuses on tagging a restricted set of light verbs in the corpus and lacks detailed representations of LVCs within specific linguistic contexts.

The rest of the paper is structured as follows. Section 2 examines linguistics properties of Chinese LVCs within the UMR framework. Section 3 introduces refined criteria for systematically identifying these constructions. Section 4 highlights the distinctions between Chinese LVCs and causative constructions. Finally, Section 5 concludes the paper by summarizing the key findings and suggesting directions for future research. These contributions aim to improve UMR annotation practices and deepen the understanding of Chinese LVCs in semantic representation frameworks.

2 Linguistic Properties of LVCs

In this section, we set aside highly grammaticalized light verbs, such as the “do” and “give” groups, to focus on syntactic and semantic structure of vague cut cases of LVCs in Chinese and examine their diverse patterns.

2.1 Argument Structure

In UMR, light verbs are treated as having zero arguments, similar to auxiliary verbs, which also lack an argument structure (Xue and Palmer, 2005). The primary function of light verbs is to provide grammatical or aspectual support to the nominal predicate, which holds the core semantic content and carries the associated arguments (Bonn et al., 2023). The argument structure of an LVC thus depends entirely on the nominal predicate, which can have zero, one, or multiple arguments.

In Chinese, some nominal predicates naturally occur without requiring main arguments. For example, the nominal predicate 爆炸 [baozha, "explode"] in sentence (2) can appear alone or together with the verb 发生 [fasheng, "occur"]. When 发生 is used, it explicitly indicates that the explosion event took place, allowing the introduction of specific details such as the time and location of the event. However, adding or omitting 发生 [fasheng, "occur"] does not change the fundamental meaning of the sentence: either way, the proposition remains that an explosion took place at the concert. Thus, 发生 [fasheng, "occur"] is considered a "light verb," as it does not contribute substantial new propositional content beyond signaling the occurrence of an event.

(2) 演唱会 于22时33分 发生-爆炸 。
concert at 22:33 occur-explode .
"An explosion occurred at the concert at 22:33."

```
(s2x / 爆炸-01 [baozha, "explode"]
:place (s2x2 / 演唱会 [yanchanghui, "concert"])
:temporal (s2d / date-entity
        :time "h22m33")
:Aspect Performance
:MODSTR FullAff
```

Certain nominal predicates involve exactly one semantic argument. Example (3) illustrates this clearly:

(3) 该团队 率先 取得-胜利 。
the team take the lead get-success .
"The team was the first to achieve victory."

```
(s3x / 胜利-01 [shengli, "success"]
  :ARG0 (s3x2 / 团队 [tuandui, "team"]
          :mod (s3x3 / 该 [gai, "the"]))
  :mod (s3x4 / 率先 [shuaixian, "take the lead"])
  :Aspect Performance
  :MODSTR FullAff)
```

The nominal predicate 胜利 [shengli, "success"] inherently involves one argument—the entity experiencing or achieving success (the team). The accompanying light verb 取得 [qude, "get"] does not introduce any additional arguments; it merely serves as a grammatical connector that enhances fluency. The UMR clearly annotates the team as ARG0, underscoring that the nominal predicate's single argument structure is preserved while the light verb remains semantically redundant.

Other nominal predicates can take multiple semantic arguments. Consider example (4):

(4) 科学家 对 遗骸 进行-检查。
scientists towards remains proceed-exam
"Scientists conducted an examination of the remains."

```
(s4x / 检查-01 [jiancha, "exam"]
  :ARG0 (s4x3 / 科学家 [kexuejia, "scientists"]
  :ARG1 (s4x5 / 遗骸 [yihai, "remains"]
  :Aspect Performance
  :MODSTR FullAff)
```

In this example, the nominal predicate 检查 [jiancha, "exam"] requires two semantic arguments: the agent performing the action (科学家 [kexuejia, "scientists"]) and the object of the action (遗骸 [yihai, "remains"]). The light verb 进行 [jinxing, "proceed"] contributes no independent semantic argument structure.

2.2 Adverbial and Attributive Modification

The incorporation of modifiers into Chinese LVCs significantly increases their structural complexity. Because Chinese adjectives can serve either attributively or adverbially, two distinct modification patterns emerge within LVCs. A modifier may directly describe the nominal predicate alone (attributive modification), or it may adverbially modify the entire LVC, thereby altering the interpretation of the entire event.

However, the possibility for a modifier to extend its scope beyond the nominal predicate should not be considered a definitive criterion in determining whether a construction qualifies as an LVC. Consider Example (5):

(5) 他们 展开 了 激烈的 争吵 。
They engage in PF intense dispute .
"They engaged in an intense dispute."

```
(s5x / 争吵-01 [zhengchao, "dispute"]
  :ARG0 (s5p / person
          :refer-person 3rd
          :refer-number Plural)
  :manner (s5x5 / 激烈 [jilie, "intense"])
  :Aspect Performance
  :MODSTR FullAff)
```

The adjective 激烈 [jilie, “intense”] allows two possible interpretations: it either describes only the nominal predicate 争吵 [zhengchao, “dispute”], resulting in “intense dispute,” or it modifies the entire event described by 展开争吵 [zhankai-zhengchao, “engage in dispute”], producing the reading “engaged intensely in a dispute).” Regardless of this ambiguity in interpretation, the argument structure remains stable, governed solely by the nominal predicate 争吵 [zhengchao, “dispute”], while the light verb 展开 [zhankai, “engage”] does not introduce any additional arguments.

Modifier placement further complicates the syntax of LVCs. Modifiers need not remain adjacent to their modified element; rather, they can freely occur either before, within, or after the construction. Example (6) illustrates a temporal modifier placed at the end of an LVC:

```
(6) 该团队 对其 进行-研究      长达 十年。
    the team on it conduct research long  ten years
    “The team conducted research on it for as long
    as ten years.”
```

```
  (s6x / 研究-01 [yanjiu, “research”]
    :ARG0 (s6x2 / 团队 [tuandui, “team”]
              :mod (s6x3 / 该 [gai, “the”]))
    :ARG1 (s6x4 / 其 [qi, “it”]
    :duration (s6t / temporal-quantity
                     :mod (s6x6 / 长 [chang, “long”])
                     :quant 10
                     :unit (s6x7 / 年 [nian, “year”]))
    :Aspect Process
    :MODSTR FullAff)
```

The temporal modifier 十年 [shinian, “ten years”] appears at the end of the construction yet semantically specifies the duration of the nominal predicate 研究 [yanjiu, “research”]. Despite this non-adjacent surface placement, the UMR annotation maintains consistency by explicitly linking this temporal element directly to the nominal predicate, highlighting the event’s duration rather than completion.

Reflexive modifiers introduce additional layers of interpretation complexity. In Examples (7) and (8), the reflexive 他们自己 [tamen-ziji, “themselves”] demonstrates ambiguity contingent upon syntactic placement. When the reflexive modifier follows the nominal predicate, as in (7), it clearly signals possession:

```
(7) 工人们 取得   了 他们自己的 胜利。
    workers achieve PF their-own    victory
    “The workers achieved their own victory.”
```

```
  (s7x / 胜利-01 [shengli, “victory”]
    :ARG0 (s7x2 / 工人 [gongren, “workers”]
              :refer-number Plural)
    :poss-of (s7x3 / 他们自己 “tamen-ziji, themselves” )
    :Aspect Performance
    :MODSTR FullAff)
```

However, placing the reflexive before the entire LVC, as in (8), conveys collective agency rather than possession.

```
(8) 工人们 他们自己 取得   了 胜利。
    workers themselves achieve PF victory
    “The workers themselves achieved victory.”
```

```
  (s8x / 胜利-01 [shengli, “victory”]
    :ARG0 (s8x2 / 工人 [gongren, “workers”]
              :refer-number Plural)
    :Aspect Performance
    :MODSTR FullAff)
```

2.3 Inherent Aspect

Events expressed through nominal constructions often pose significant challenges for aspectual annotation, primarily because they lack the explicit morphological or syntactic markers that typically signal aspectual distinctions. In LVCs, light verbs frequently combine with nominal predicates, thereby clarifying or altering the aspectual interpretation. While the default assumption might be that the aspect of a light verb aligns seamlessly with the nominal event it accompanies, discrepancies can occur and warrant careful analysis. For instance, in many instances, the light verb and the nominal share the same aspectual value, as exemplified by (10), where both 给予 [jiyu, “offer] and 帮助 [bangzhu, “assist”] converge on a process aspect indicating ongoing activity of offering help. Such compatibility between the lexical aspect of the light verbs and the nominal predicate typically simplifies the annotation process because it provides a clear indication that the event has a definite end point.

(9) 慈善机构 承诺 给予-帮助。
charity organization promise offer-assist
"The charity organization promised to offer assistance."

```
(s10x / 承诺-01 [chengnuo, "promise"]
  :ARG0 (s10x2 / 机构 [jigou, "organization"]
       :mod (s10x4 / 慈善 [cishan, "charity"]))
  :ARG1 (s10x3 / 帮助-01 [bangzhu, "assist"]
       :Aspect Process
       :MODSTR FullAff)
  :Aspect Performance
  :MODSTR FullAff)
```

However, more nuanced cases arise when the inherent aspect of the light verb diverges from that of the nominal event. Example (10) illustrates this situation: the nominal event 会谈 [huitan, "meet"] is intrinsically durational, unfolding over a three-hour span, thus suggesting an ongoing process. By contrast, the light verb 举行 [juxing, "hold"] tends to convey a sense of a discrete and complete occurrence—what can be classified as a performance aspect in UMR. When these two aspectual profiles come together in an LVC, the annotation must account for the fact that the event unfolds over a span of time but also concludes definitively once the meeting has taken place.

(10) 双方 举行 三小时 会谈。
both sides hold 3 hours meet
"Both sides held a three-hour meeting."

```
(s11x / 会谈-01 [huitan, "meet"]
  :ARG0 (s11x2 / 双方 [shuangfang, "both sides"]
  :temporal (s11t / temporal-quantity
         :quant 3
         :unit (s11x3 / 小时 [xiaoshi, "hour"]))
  :Aspect Performance
  :MODSTR FullAff)
```

2.4 Existential and Passive Oriented

In Chinese, there are two special sentence patterns that closely relate to LVCs: those oriented toward existence and those oriented toward passivity. The first type includes examples with the verbs 有 [you, "have"] or 存在 [cunzai, "exist"], both of which can function as light verbs in specific contexts. It is important to distinguish these uses from the typical Chinese existential you-construction, which parallels the English there-construction and expresses the existence, appearance, or disappearance of entities at a particular place or time. Consider the example in (11): although 存在 [cunzai, "exist"] typically means 'exist,' it does not convey its usual existential meaning but instead serves as a light verb. In this usage, it indicates a static relational state between the arguments rather than literal existence. The lexical meaning of 存在 [cunzai, "exist"] is bleached, and it instead operates as a grammatical marker that highlights the aspectual or stative nature of the nominal predicate.

(11) 他 与 袭击案 存在-关联 。
he with the attack case exist-connect .
"He is connected to the attack case."

```
(s13x / 关联-01 [guanlian, "connect"]
  :ARG0 (s13p / person
       :refer-person 3rd
       :refer-number Singular)
  :ARG1 (s13x2 / 袭击案 [xiji-an, "attack case"])
  :Aspect State)
```

The second category involves passive-oriented syntactic patterns, where the grammatical structure shifts from an active to a passive voice while preserving the core semantic representation. Crucially, this syntactic alternation—exemplified by passive markers such as 受到 [shoudao, "undergo"], 遭到 [zaodao, "suffer"], and 被 [bei, "be"]—does not alter the thematic roles or event structure of the nominal predicate. Light verbs act as voice heads that license syntactic reconfiguration without modifying lexical-semantic content. In (12), while the passive construction elevates the patient 敌人 [diren, "enemy"] to subject position, the nominal predicate 攻击 [gongji, "attack"] remains the semantic core of the event. The light verb 受到 [shoudao, "undergo"] functions solely to signal passive voice and suppress roles.

(12) 敌人 受到 猛烈 攻击。
enemy undergo fierce attack
"The enemy was heavily attacked."

```
(s15x / 攻击-01 [gongji, "attack"]
```

:ARG1 (s15x2 / 敌人 [diren, “enemy”])
:manner (s15x3 / 猛烈 [menglie, “fierce”])
:Aspect Performance
:MODSTR FullAff)

3 Broad criteria for determining LVCs

Building on the tests developed by PARSEME, the UMR annotation guidelines for LVCs in Chinese introduce four specific tests to determine whether a verb with a predicative noun as complement qualifies as an LVC.

3.1 Test 1

Test 1 evaluates whether the complement of a light verb is a predicative noun. For example, in the phrase “make a contribution,” the noun “contribution” is predicative because it represents an event or action that corresponds to the verb “contribute.” Conversely, in “make a cake,” the noun “cake” is not predicative, as it does not have a verbal counterpart. Therefore, the former passes Test 1 and proceeds to the next stage, while the latter is excluded.

Notably, verbs are often mistaken for predicative nouns in Chinese, primarily because of the unmarked morphological status shared by predicative nouns and their verbal counterparts. For instance, in (13), the verb combination 引用报导 specifically conveys the meaning “to be cited and reported,” rather than suggesting that Chinese media cited reports or the reporting event created by other outlets. In the latter interpretation, 报导 [baodao, “report”] would act as a predicative noun. However, in this context, it does not meet the requirements of Test 1, as it functions as a verb.

(13) 中文 媒体 引用 报导 该 新闻。
Chinese media cite report the news
“The Chinese media cited and reported the news.”

(s16a / and
:op1 (s16x / 引用-01 [yinyong, “cite”]
:ARG0 (s16x2 / 媒体 [meiti, “media”]
:medium (s16x3 / 中文 [zhongwen, “Chinese”]))
:ARG1 (s16x4 / 新闻 [xinwen, “news”]
:mod (s16x6 / 该 [gai, “the”])))
:Aspect Performance
:MODSTR FullAff)
:op2 (s16x7 / 报导-01 [baodao, “report”]
:ARG0 s16x2
:ARG1 s16x4
:Aspect Performance
:MODSTR FullAff))

3.2 Test 2

Test 2 assesses whether the subject of a verb within the construction also functions as an argument of the nominal predicate. For instance, in the sentence “made a presentation to his boss,” the subject of the verb “make” serves as the agent of the nominal predicate “presentation,” thereby satisfying Test 2. Conversely, in “John’s boss interrupted his presentation,” the subject “John’s boss” does not hold a thematic role related to the nominal predicate “presentation,” resulting in a failure to meet Test 2.

In most cases, constructions that pass Test 1 also pass Test 2. However, exceptions do exist. For instance, in (14), the verb 支持 [zhichi, “support”] in the expression 支持反恐 [zhichi-fankong, “support counter-terrorism”] presents a counterexample. In this case, the subject of 支持 [zhichi, “support”] is not inherently an argument of the nominal predicate 反恐 [fankong, “counter-terrorism”], as it does not directly engage in anti-terrorism actions. Rather, the subject expresses an external stance of approval or endorsement, without participating in the actual event, thus failing Test 2.

(14) 清真寺 曾 就 支持 反恐
Mosque once on support counter-terrorism
和 生命尊严 布告。
and life dignity issue a statement
“The mosque once issued a statement supporting counter-terrorism and the dignity of life.”

(s17x / 布告-00 [bugao, “issue a statement”]
:ARG0 (s17x2 / 清真寺 [qingzhensi, “Mosque”])
:ARG1 (s17a / and
:op1 (s17x4 / 支持-01 [zhichi, “support”]
:ARG1 (s17x5 / 反-01 [fan, “counter”]
:ARG1 (s17x6 / 恐 [kong, “terrorism”])))
:op2 (s17x7 / 尊严 [zunyan, “dignity”]

```
            :mod (s17x8 / 生命 [shengming, “life”])))
    :mod (s17x3 / 曾 [ceng, “once”])
    :Aspect Performance
    :MODSTR FullAff)
```

3.3 Test 3

Test 3 is designed to determine whether a given verb introduces substantial semantic content beyond merely hosting morphological features such as tense, mood, and aspect, or contributing syntactic structure for the nominal predicate. In essence, this test seeks to distinguish genuinely “light” verbs from those that add meaningful lexical semantics. If a verb simply facilitates the nominal predicate’s argument structure or supplies grammatical inflections without introducing new propositional content, it can be considered light.

In UMR, applying Test 3 is relatively straightforward. If removing the verb does not alter the core propositional meaning, the verb can be classified as light. However, if the omission leads to a loss or shift in essential semantic content, the verb is not considered light. For example, in (15), the verb 引起 [yinqi, “draw”] contributes more than just grammatical support—it introduces the causative meaning of bringing about attention. This is semantically richer than the role of a typical light verb, which would merely provide aspectual or syntactic support to the nominal predicate 注意 [zhuyi, “attention”] without adding new event content. Thus, 引起 [yinqi, “draw”] fails the test for lightness, as it adds distinct lexical meaning to the clause.

(15) 他 已 引起 情报部门
he already draw the intelligence agency
注意 。
attention.
“He has drawn the attention of the intelligence department.”

```
(s19x / 引起-01 [yinqi, “draw”]
    :ARG0 (s19p / person
        :refer-person 3rd
        :refer-number Singular)
    :ARG1 (s19x2 / 注意-01 [zhuyi, “attention”]
        :ARG0 (s19x3 / 部门 [bumen, “agency”]
            :mod (s19x4 / 情报 [qingbao, “intelligence”]))
        :Aspect State
        :MODSTR FullAff)
    :mod (s19x5 / 已 [yi, “already”])
    :Aspect Performance
    :MODSTR FullAff)
```

4 Causative Constructions

Certain verb constructions in Chinese resemble LVCs in form or function, particularly those expressing causative relations, but they do not fully satisfy the core definitional criteria of LVCs. While not the primary focus of this study, these constructions merit careful consideration, as their syntactic and semantic characteristics can easily be mistaken for genuine LVCs.

In certain complex transitive verb constructions that can be interpreted as externally caused events, the process of causativization consistently appears to be feasible. Basciano (2013), for instance, highlights verbs such as 弄醒 [nongxing, “wake”), 弄哭 [nongku, “make cry”], 搞丢 [gaodiu, “lose”], and 搞坏 [gaohuai, “destroy”), all of which demonstrate this pattern. Similarly, Chung (2006) investigates verbs containing the root 加 [jia, “add”], including 加宽 [jiakuan, “widen”], 加高 [jiagao, “heighten”], and 加强 [jiaqiang, “to strengthen”], observing that the 加 [jia, “add”] facilitates the formation of transitive counterparts of change-of-state verbs derived from open-scale adjectives that denote incremental increases.

Causative constructions inherently involve distinct semantic roles for both the causing event and the resultant state (Tham, 2015; Sun et al., 2023). Therefore, the constituent verbs in such constructions function as independent predicates, each maintaining its own argument structure. This property distinguishes causative verb compounds clearly from true LVCs, even if one verb appears semantically “lighter” than the other. Specifically, the so-called “light” verb in causative compounds still contributes a distinct argument structure, disqualifying it from classification as a genuine light verb. For example, in (16), the verb compound 打破 [dapo, “break”] encodes two separate events: an action event 打 [da, “beat”] and a resultant state 破 [po, “break”]. Thus, 打 [da, “beat”] denotes a causing action, and 破 [po, “break”] expresses the resulting event. This forms a compositional resultative construction rather than an LVC.

(16) 他 打-破 了 桌上的 花瓶。
he beat-break PF on the table vase
“He broke the vase on the table.”

```
(s21x / 打-015 [da, “beat”]
  :ARG0 (s21p / person
        :refer-person 3rd
        :refer-number Singular)
  :Cause-of (s21x2 / 破-04 [po, “break”]
            :ARG0 (s21x3 / 花瓶 [heaping, “vase”]
                  :place (s21x4 / 桌上 [zhuozi, “table”))
            :Aspect State
            :MODSTR FullAff)
  :Aspect Performance
  :MODSTR FullAff)
```

From a semantic perspective, metaphorization involves extending a verb’s literal, physical meaning into a more abstract domain. In Chinese, certain verbs display such metaphorization, making their identification and annotation more challenging. The same verb compound 打破 [dapo, “break”] can function in both a literal sense, as seen in (16), and a metaphorical sense in (17), where 打破 [dapo, “break”] is best interpreted as expressing the disruption or alteration of an abstract state. In UMR annotation, the first frame of 打破 [dapo, “break”] treats the bird sound (ARG0) as the agent and the stillness (ARG1) as the theme.

(17) 鸟声 打破 了 清晨的宁静。
the sounds of birds break PF tranquility of the early morning
“The sound of birds broke the tranquility of the early morning.”

```
(s22x / 打破-01 [dapo, “break”]
  :ARG0 (s22x2 / 鸟声 [niaosheng, “the sound of birds”])
  :ARG1 (s22x3 / 宁静 [ningjing, “tranquility”]
        :temporal (s22x4 / 清晨 [qingchen, “the early morning”]))
  :Aspect Performance
  :MODSTR FullAff)
            :quant (s26x4 / 多)))
  :ARG1 (s26x5 / 慰问)
  :Aspect Performance
  :MODSTR FullAff)
```

5 Conclusion

In this paper, we have explored the challenges inherent in annotating Chinese light verb constructions within the Uniform Meaning Representation framework. Through an analysis of their structural and semantic characteristics, we illustrated how the syntactic flexibility of Chinese light verbs complicates accurate annotation. We addressed key issues such as identifying LVCs, annotating argument structures, and distinguishing these constructions from similar forms. Our findings reinforce prior research (Savary et al., 2017; Lin et al., 2014), confirming that Chinese light verbs primarily fulfill grammatical roles rather than contributing substantive semantic meaning. Nevertheless, their distinctive syntactic versatility calls for refined annotation guidelines to mitigate potential misclassifications. To address this, we proposed a dual-path annotation method, separately encoding the argument structure of nominal predicates and the grammatical properties of light verbs. This methodology sets the stage for future studies to investigate the intricate syntactic, semantic, and contextual dimensions of LVCs. Ultimately, our work aims to enhance both linguistic research and computational modeling of Chinese and other languages exhibiting similar complexities.

Acknowledgments

This work was supported by a grant from the CNS Division of the National Science Foundation (Award No. 2213804), titled “Building a Broad Infrastructure for Uniform Meaning Representations.” Additional financial support was provided by the Guangdong Provincial Philosophy and Social Science Planning General Project (Grant No. GD23CWY10) and a research grant (No. LAI202302) from the Laboratory of Language and Artificial Intelligence, Center for Linguistics and Applied Linguistics, Guangdong University of Foreign Studies. The views expressed in this paper are those of the authors and do not necessarily reflect those of the National Science Foundation, the Guangdong Provincial Philosophy and Social Science Planning Office, or the Laboratory of Language and Artificial Intelligence. We are grateful to William Croft, James Pustejovsky, Martha Palmer, Andrew Cowell, and other collaborators on the UMR project for their insightful discussions. We also thank the annotators of the Chinese UMR team, particularly Haibo Sun, Hsiao-Jung Cheng, and Dan Xing for raising important issues related to the annotation of Chinese Light Verb Constructions. All errors and omissions remain the responsibility of the authors.

References

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., & Flickinger, D. 2002. Multiword expressions: A pain in the neck for NLP. In Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002, Mexico City (pp. 1–15).

Stevenson, S., Fazly, A., & North, R. 2004. Statistical measures of the semi-productivity of light verb constructions. In Proceedings of the Workshop on Multiword Expressions: Integrating Processing (pp. 1–8).

Tu, Y., & Roth, D. 2011. Learning English light verb constructions: Contextual or statistical. In Proceedings of the Workshop on Multiword Expressions: From Parsing and Generation to the Real World (pp. 31–39).

Vincze, V., Nagy, I., & Berend, G. 2011. Detecting noun compounds and light verb constructions: A contrastive study. In Proceedings of the Workshop on Multiword Expressions: From Parsing and Generation to the Real World (pp. 116–121).

Nagy, I., Rácz, A., & Vincze, V. 2020. Detecting light verb constructions across languages. Natural Language Engineering, 26(3), 319–348.

Savary, A., Ramisch, C., Cordeiro, S. R., Sangati, F., Vincze, V., Qasemi Zadeh, B., Candito, M., Cap, F., Giouli, V., Stoyanova, I., & Doucet, A. 2017. The PARSEME shared task on automatic identification of verbal multiword expressions. In Proceedings of the 13th Workshop on Multiword Expressions at EACL (pp. 31–47).

Bonn, J., Cowell, A., Hajic, J., Palmer, A., Palmer, M., Pustejovsky, J., Sun, H., Uresova, Z., Wein, S., Xue, N., & Jin, Z. 2023. UMR annotation of multiword expressions. In Proceedings of the 4th International Workshop on Designing Meaning Representations.

Butt, M. 2010. The light verb jungle: Still hacking away. In Complex Predicates in Cross-Linguistic Perspective (pp. 48–78).

Lin, J., et al. 2014. Annotation and classification of light verbs and light verb variations in Mandarin Chinese. In Proceedings of the Workshop on Lexical and Grammatical Resources for Language Processing.

Huang, C. R., Lin, J., Jiang, M., & Xu, H. 2014. Corpus-based study and identification of Mandarin Chinese light verb variations. In Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects (pp. 1–10).

Jiang, M., Klyueva, N., Xu, H., & Huang, C. R. 2018. Annotating Chinese light verb constructions according to PARSEME guidelines.

Sun, H., Zhu, Y., Zhao, J., & Xue, N. 2023. UMR annotation of Chinese verb compounds and related constructions. In Proceedings of the First International Workshop on Construction Grammars and NLP (pp. 75–84).

Bonn, J., Buchholz, M. J., Chun, J., Cowell, A., Croft, W., Denk, L., Ge, S., Hajic, J., Lai, K., Martin, J. H., & Myers, S. 2024. Building a broad infrastructure for Uniform Meaning Representations. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) (pp. 2537–2547).

Xue, N., & Palmer, M. 2005. Automatic semantic role labeling for Chinese verbs. In Proceedings of IJCAI (pp. 1160–1165).

Tsou, B. K., & Yip, K. F. 2020. A corpus-based comparative study of light verbs in three Chinese speech communities. In Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation (pp. 302–311).

Lu, L. 2016. Information structure in Mandarin Chinese light verb constructions. SOAS Working Papers in Linguistics, 18, 53–74.

Wang, D., Jiang, G., & Zheng, Y. 2023. Walking out of the light verb jungle: Exploring the translation strategies of light verb constructions in Chinese–English consecutive interpreting. Frontiers in Psychology, 14, 1113973.

Basciano, B. 2013. Causative light verbs in Mandarin Chinese (and beyond). In Morphology in Toulouse: Selected Proceedings of Décembrettes, 7(1), 57–89.

Chung, K. S. 2006. Mandarin compound verbs. Crane Publishing.

Tham, S. W. 2015. Resultative verb compounds in Mandarin. In The Oxford Handbook of Chinese Linguistics (pp. 306–321).

Universal Dependencies for the Alemannic Alsatian Dialects

Barbara Hoff, Nathanaël Beiner, Delphine Bernhard
Université de Strasbourg, LiLPa UR 1339
F-67000 Strasbourg, France
{barbara.hoff,n.beiner,dbernhard}@unistra.fr

Abstract

We present the first corpus of Alsatian Alemannic dialects following Universal Dependencies (UD) guidelines, a project which already covers many of the world's languages. Standard languages are represented to a greater extent than non-standard varieties in UD, and our corpus contributes to closing the gap in the lack of resources for Alsatian dialects by presenting the first UD treebank for these dialects, which are spoken in Northeastern France. Our corpus is annotated both with part-of-speech tags and dependency information, as well as French glosses and German lemmas, containing in total 975 sentences and 19,286 tokens, spanning over various text genres. In this article, we present our data, details of the annotation process, as well as some specific syntactic phenomena which differentiate and situate Alsatian with regards to both Standard German and some other German non-standard varieties. The addition of this corpus to the UD project allows for a higher visibility of the Alemannic Alsatian dialects in linguistic research, and provides a valuable resource for research in many fields, including NLP, syntax and comparative Germanic linguistics.

1 Introduction

The project of Universal Dependencies (UD) (Zeman et al., 2024) has the goal of providing cross-linguistic annotation guidelines and treebanks for linguistic research. As of March 2025, there are 296 treebanks in 168 languages available as part of the UD project. While some non-standard German and Germanic varieties are represented (see Section 2), there is currently no UD treebank for Alsatian, an Upper German dialect spoken in the east of France. To address this gap in research, we present the first UD treebank of Alemannic Alsatian dialects, ranging over different source texts and genres. This article will present our work and our corpus, and describe annotation guidelines for some phenomena found in Alsatian. We will first provide a background for the Alemannic Alsatian dialects and available UD resources for non-standard German varieties (Section 2). We will then provide more information about our data and the annotation process (Section 3), and provide examples of some syntactic phenomena in Alsatian and related annotation decisions (Section 3.3 and 3.4).

2 Alsatian and Related Languages in UD

The terms 'Alsatian', in Alsatian 'Elsässisch, Elsassisch', marginally 'Elsässerdytsch', and in French 'alsacien' are hypernyms which refer to all the Alemannic and Franconian dialectal varieties spoken in the Alsace region in Northeastern France. These terms are used by the Alsatian population itself, whether they speak Alsatian or not.

There is no widely used written standard for Alsatian. Various spelling systems have been developed and proposed, to make it possible for all speakers to write in their own variety of Alsatian with shared grapheme to phoneme rules. However, most speakers are not familiar with these spelling systems, and there is thus a lot of variation in how speakers write Alsatian, depending both on the specific variety they speak and on the degree of influence of French and Standard German spelling (Beiner, 2022).

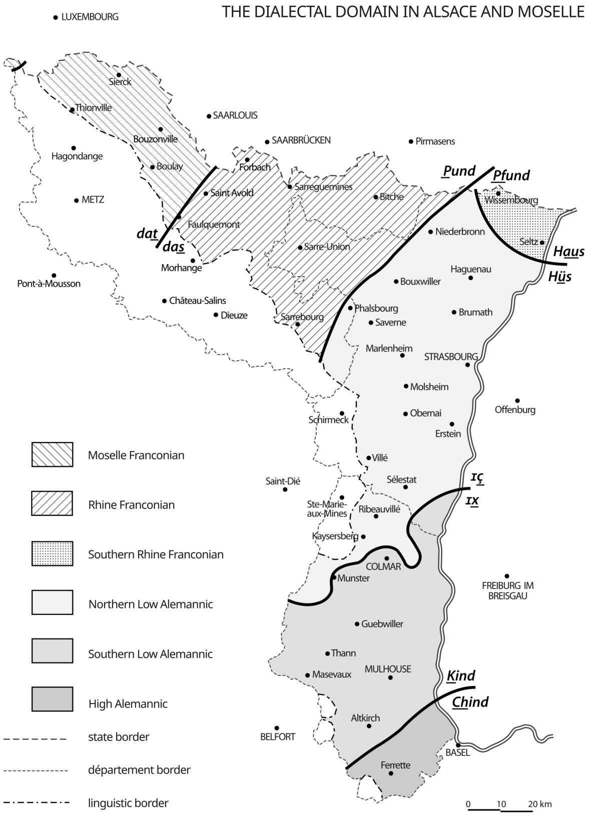
The linguistic terms used to refer to this group of dialectal varieties, e.g. 'Rhine Franconian' or 'Low Alemannic', have been chosen by linguists in reference to the Alemanni and Frankish tribes who settled in Alsace and in the surrounding areas during the 5th century and who were speaking Germanic languages. It is the descendants of those Germanic varieties which are still spoken in Alsace nowadays, as well as in the surrounding regions in Germany and Switzerland.

As shown in Figure 1, the different Upper Ger-

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 10–22
August 28-29, 2025 ©2025 Association for Computational Linguistics

man dialects spoken in Alsace are:

- **Rhine Franconian**, characterised by the use of *Pund, Appel* [p^{h}ʊnd̥], [ɒb̥l̥̩] instead of *Pfund, Apfel* [b̥fʊnd̥], [ɒb̥fl̥̩] in the southeast of the isophone,
- **South Rhine Franconian**, with the forms *Haus, Ais* [haws, ajs] (New High German diphthongisation) instead of *Hüs, Ys* [hys, is] kept as monophthongs in the south,
- **Northern and Southern Low Alemannic**, differentiated from one another by the ich-laut pronunciation, respectively [ç] or [ʃ] after front vowels and [x] after back vowels in Northern Low Alemannic, and [x] in all positions in Southern Low Alemannic (e.g. *ich, Büch, fycht* [ɪç, b̥yç, fiçd̥ – ɪx, b̥yx, fixd̥]),
- and **High Alemannic**, with *Chind, Chatz* [xɪnd̥, xɒd̥s] instead of *Kind, Katz* [k^{h}ɪnd̥, k^{h}ɒd̥s] in the north.



Map created by Anne Horrenberger for the University of Strasbourg (2017)

Figure 1: The dialectal domain in Alsace and Moselle.

They are currently four Standard German treebanks annotated with UD dependencies: HDT (Borges Völker et al., 2019), GSD (McDonald et al., 2013), PUD (Zeman et al., 2017) and LIT (Salomoni, 2017). In addition, there are four treebanks of High and Low German varieties: Bavarian (Blaschke et al., 2024), Swiss German (Aepli and Clematide, 2018), Luxembourgish (Plum et al., 2024), and Low Saxon (Siewert and Rueter, 2024).

We present the first corpus of texts in the Alemannic Alsatian dialects annotated following UD guidelines. It is the second corpus for Alemannic in general, after the Swiss German corpus.

3 Corpus Data

The data for our corpus comes from different sources, spanning different genres of texts, as summarised in Table 1. The training corpus consisted of texts T1–T4.[1] Most texts are in Northern Low Alemannic, except for T3 and 13, as well as some sentences in T12, T14, T15, and T17, which are in Southern Low Alemannic. There are also some sentences in High Alemannic in T14. T5 to T11 are complete texts, while the other texts are excerpts. In total, the corpus contains 975 sentences and 19,286 tokens.

Texts annotated in phase 2 are all translations, usually from French, and sourced from the ParCoLab parallel corpus (Stosic et al., 2024). Some of the texts in the corpus were professionally translated, while others were translated by a member of the project. Both before and during the annotation process, annotators identified and discussed germanisms in the texts, i.e. forms influenced by Standard German which are not part of the traditional Alsatian norm, but that can be used by some speakers nowadays in what we call the modern Alsatian norm. In the texts annotated by a member of the project, such forms were replaced with an Alsatian form judged more accurate with regards to traditional use, and it is the corrected sentence which was annotated. In texts which were professionally translated and transcripts of spoken language, the original version was kept in the `#text_origin` metadata and the corrected sentence was indicated in the `#text` metadata, as was done in the Low Saxon corpus (Siewert and Rueter, 2024, p. 15,977).

[1] The source text of examples used in this article is indicated by the text identifier (T1–11) and the number of the sentence (for example, *s38* for ‘sentence 38’) according to the numbering of sentences in our corpus. When there is no relevant example available in the corpus, we used constructed examples and do not indicate a text identifier. For Text 11 (T11), we further specify the date of the specific chronicle we want to refer to.

Text	Text and author	Genre	Sentences	Tokens
Phase 1: Training – double annotation				
T1	D’r Hoflieferant, Gustave Stoskopf	📕 Fiction – Theatre	13	88
T2	Recipes, Office pour la Langue et les Cultures d’Alsace et de Moselle (OLCA)	🎂 Cooking – Recipe	17	170
T3	Haut-Rhin Magazine	ℹ Non-Fiction – Journalistic	15	121
T4	Alemannic Wikipedia	W Wiki	17	188
Phase 2: Translated texts – double annotation				
T5	Monday Tales, Alphonse Daudet	📕 Fiction	179	3,804
T6	The Universal Declaration of Human Rights	🔨 Legal	83	2,183
T7	The Decameron, Boccaccio	📕 Fiction	19	483
T8	Pierre and the Wolf, Sergueï Prokofiev	📕 Fiction	65	925
T9	The Prodigal Son, Luke (Steiner and Matzen, 2016)	☁ Fiction – Bible	29	628
T10	The North Wind and the Sun, Aesop (Boula de Mareüil et al., 2018)	📕 Fiction	6	126
T11	Chronicles about the regional languages of France, Michel Feltin-Palas	ℹ Non-Fiction – Journalistic	177	4,267
Phase 3: Selected sentences – single annotation				
T12	Linguistic and ethnographic atlas of Alsace	💬 Spoken, Ethnotext	60	2,711
T13	Haut-Rhin Magazine	ℹ Non-Fiction – Journalistic	12	249
T14	Miscellaneous literary texts, various authors	📕 Fiction	33	370
T15	Miscellaneous texts, OLCA	📕 Fiction, ℹ Non-Fiction, 🎂 Cooking	19	163
T16	Miscellaneous theatre plays, some translated from French, various authors	📕 Fiction – Theatre	171	1,528
T17	Alemannic Wikipedia	W Wiki	60	1,282

Table 1: Source texts, genre distribution, and number of surface tokens and sentences per text in the corpus. Texts T1-T4 were only included in the training batch.

3.1 Tokenisation

The corpus was tokenised using an adapted version of the tokenisation script developed for Bavarian (Blaschke et al., 2024).

The tokenisation was manually checked by the annotators in order to, for example, split contracted forms of a preposition and a determiner (*im* into *i* + *m* ’in the’) and to correct mistakes when sentences were wrongly split or merged together. Following German UD rules, as well as annotation decisions for Bavarian (Blaschke et al., 2024, p. 10924) and Luxembourgish (Plum et al., 2024, p. 32), contracted forms consisting of a preposition and a determiner were split (see example above), while hyphenated compounds were not (ex: *Grìeni-Lìnse* ‘green lentils’ (T2, s9)). Epenthetic consonants and associated punctuation were not split, but merged with the previous word, see section 3.4 for details about their annotation.

3.2 Annotation Procedure

The annotators always worked by correcting automatic pre-annotations in order to streamline the annotation process. In phases 1 and 2, the corpus was pre-annotated using three main methods: UDPipe (Straka, 2018), Mistral Large with prompts,[2] and the trainable parsing service on the Arborator-Grew platform (Guibon et al., 2020).[3] See Bernhard et al. (2025) for more information about the pre-annotation process for this corpus.

In phase 3, the annotations from phrase 1 and phase 2 were used to train a new pre-annotation model using the MaChAmp toolkit (van der Goot et al., 2021). In order to increase the amount of available training data, we also used the test splits of the following existing UD 2.15 corpora: German GSD (McDonald et al., 2013), Bavarian MaiBaam (Blaschke et al., 2024), Swiss German UZH (Aepli and Clematide, 2018) and Luxembourgish LuxBank (Plum et al., 2024). Training targeted the following tasks: part-of-speech (POS), dependency relations and German lemma (when available in the training corpus). The model was then used to obtain pre-annotations for sentences from varied source material (see Table 1, Phase 3). We then selected the sentences to be corrected by the annotators using uncertainty sampling based

[2] https://mistral.ai/fr/news/mistral-large-2407

[3] In phase 1, GPT 3.5 and 4 were also used, as well as different training corpora for ArboratoGrew.

on the label probability for POS, head and dependency relation of MaChAmp's predictions. We retained sentences with the largest average uncertainty and with at least three tokens. The objective here was to annotate sentences with phenomena that are more difficult to annotate automatically.

The annotators were two recently graduated master's students with a background in linguistics, both native speakers of French and Northern Low Alemannic and co-authors of this paper. They were hired and compensated according to local standards.

In phase 1, the annotators were trained on a small corpus (the training batch, T1–T4) to familiarise them to Universal Dependencies guidelines. The corpus was divided into the following batches:

- Phase 1 : 1 batch, with double annotations
- Phase 2 : 6 batches, with double annotation
- Phase 3: 2 batches, with a single annotation each

Dividing the corpus into different batches made it possible to quality check more effectively after discussing annotations for a batch, as well as update the annotation guidelines and correct the annotations done so far. During the annotation process in phase 1 and phase 2, the annotators only had access to their own annotations (blind annotation), in order to minimise bias. The corpus was annotated using the ArboratorGrew platform (Guibon et al., 2020) and meetings were regularly scheduled to discuss disagreements in annotation and difficult cases, and to agree on a final version for the batches with double annotation. Annotation decisions were based on UD guidelines for German, as well as annotation decisions in related varieties, like Bavarian (Blaschke et al., 2024) and Swiss German (Aepli and Clematide, 2018). GrewMatch (Guillaume, 2021) was used to access the relevant treebanks. The UD validation script[4] integrated to ArboratorGrew was used to detect mistakes. The inter-annotator agreements were high for phase 2: POS $\kappa \geq 0.90$, dependency $\alpha \geq 0.88$ (Bernhard et al., 2025).

A GitLab repository was used for storing the annotations after each step (blind annotation by each annotator, validated versions). After each upload, a verification pipeline was automatically launched to obtain a report on potential detectable errors in the annotations using Udapi (Popel et al., 2017), in particular the `MarkBugs` module. The pipeline also generated tables showing a side-by-side comparison of the annotations by the two annotators, to facilitate the identification of disagreements. Another GitLab repository was used to write the annotation guide. Each section of the annotation guide is written in a Markdown file and a pipeline based on Pandoc (MacFarlane et al., 2025) automatically generates an HTML version of the guide after each modification.

Annotation times for each batch are indicated in Table 2 for each annotator (A1[5] = annotator 1, A2 = annotator 2). The discussion time for each batch, i.e. meetings during which the two annotators compared their annotations, discussed and resolved disagreements in order to agree on a final version, varied between 5 and 10 hours. Both the annotation and discussion time for the first batches were considerably longer than later batches, since both annotators first had to familiarise themselves with UD guidelines and establish annotation rules for Alsatian. The annotation period for our corpus took place over a period of about 7 months between September 2024 and April 2025.

Batch	Sentences	Tokens	A1	A2
0	62	567	6h	5h40m
1	88	1,947	16h	12h45m
2	93	1,929	8h	7h45m
3	92	1,933	7h	6h50m
4	83	1,672	6h	6h10m
5	104	2,554	8h30m	6h15m
6	98	2,381	6h30m	5h
7	176	3,272	16h45m	/
8	179	3,031	/	17h20m
Total	975	19,286	74h45m	68h30m
Average	108	2,143	9h20m	8h33m

Table 2: Annotation time and information about the nine annotation batches

3.3 POS Tags and Dependencies

The annotation guide developed by the annotators and other members of the project is available online.[6] This section presents some annotation decisions and syntactic constructions specific to Alsatian. The Alsatian dialects differ phonetically to a large extent from standard German ('fragen'

[4] https://github.com/UniversalDependencies/tools

[5] The number used for each annotator corresponds to the number used in (Bernhard et al., 2025)

[6] French version DOI: 10.34847/nkl.0eac4288 ; English version DOI: 10.34847/nkl.5b6cs6wu.

– ‘fröje, fröwe, froja’), and to a lower extent lexically (‘Kartoffel’ – ‘Grumbeer, Aardepfel’), morphologically (e.g. reduced case system, see the annotation guide) and syntactically (see below). See Tables 4 and 3 in the appendix for details of the statistical distribution of POS tags and dependencies in the corpus.

The same POS tags and dependencies as defined in the German UD guidelines[7] were used to annotate our corpus, with the addition of a few subrelations to add more details. For example, we used the subrelations `:lmod` `:tmod` and `:emph` for the dependencies `advmod`, `obl`, and `nmod`, to indicate when the dependency provided information about location, time, or emphasis. This was not done in other Germanic corpora.

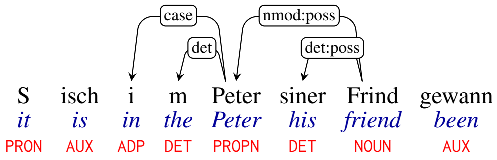
3.3.1 Noun Phrases

Possessive construction Possession is expressed in Alsatian using analytic possessive constructions instead of the genitive case. Two types of constructions are used: either using a prenominal dative, optionally reinforced with the preposition *in*, or using a prepositional phrase starting with *vun* ‘of’ following the possessed.

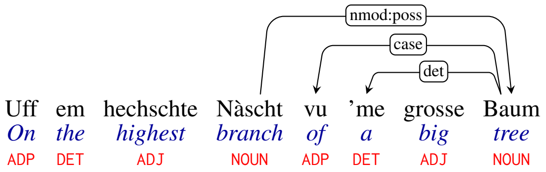
The prenominal dative structure is also found in Bavarian (Blaschke et al., 2024, p. 10926), Luxembourgish (Plum et al., 2024, p. 34) and Low Saxon (Siewert and Rueter, 2024, p. 15979).

We follow annotation guidelines for Bavarian and Luxembourgish and annotate the prenominal dative as follows: in example (1), the possessive pronoun is annotated `det:poss` of the possessed, and the possessor, `nmod:poss` of the possessed. The preposition *in* is annotated as `case` of the possessor, and the determiner, `det` of the possessor. The *vun*-construction (2) is annotated with `nmod:poss`.

(1) Here, *i* ‘in’ is a reinforcement of the dative. ‘It was Peter’s friend.’ (T8, s4)


case
det
nmod:poss
det:poss
S isch i m Peter siner Frind gewann
it is in the Peter his friend been
PRON AUX ADP DET PROPN DET NOUN AUX

(2) ‘On the highest branch of a big tree.’ (T8, s3)


nmod:poss
case
det
Uff em hechschte Nàscht vu ’me grosse Baum
On the highest branch of a big tree
ADP DET ADJ NOUN ADP DET ADJ NOUN

Dative case The preposition *in* can also be used as reinforcement of the dative case in other contexts. Below, *Heidelbeere* and *Fräu* are annotated with `obl:arg`, in the same way as dative direct objects, instead of `obl`, as prepositional phrases are usually annotated.

(3) *D’ kleine schwàrze Glickle hàn äu* ***in***
The little black eyes have also **in**
nàsse Heidelbeere geglìche.
wet blueberries resembled.
‘The small black eyes resembled also wet blueberries.’ (T5, s170)

(4) *Ich hab ’s* ***in*** *de Fräu gseit.*
I have it **to** the woman said.
Ger. ‘Ich habe es ∅ der Frau gesagt.’

Personal names with a determiner As is also the case in Bavarian (Blaschke et al., 2024, p. 10926) and Luxembourgish (Plum et al., 2024, p. 34), personal names in Alsatian can occur with a determiner. We annotate it as `det` of the name, and use `flat` for personal names with titles (*de Mösiö Hamel* ‘Mister Hamel’ (T5, s3)) and `flat:name` for personal names with first and last name (*de Laurent Lafforgue* (T11/20211123, s6)).

3.3.2 Subordinate Clauses

Relative marker Relative clauses are introduced by the relative marker *wo / wu / wü / wi* in Alsatian. These forms come from Middle High German (MHG) *wâr*[8] ‘where’ that shifted, probably very early, from /ar/ to /u/ then palatalised to /y/ in the Masevaux valley or the Kochersberg, according to Beyer (1964, p. 156). This form is invariant, and Standard German relative markers as *der, die, das* are not used in Alsatian, as is also the case in Bavarian (Blaschke et al., 2024, p. 10926) and Swiss German.

We annotate the relative marker with the POS `PRON` and with the syntactic dependency of the element it replaces. The relative marker is `obj` in example (5).

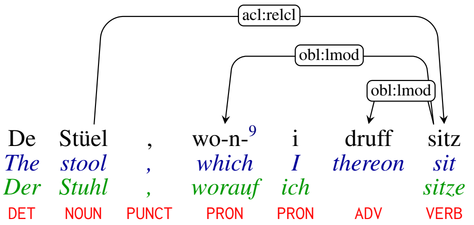
[7] https://universaldependencies.org/de/index.html

[8] *Wo / wu / wü* all come from MHG *wâr*, but the exact etymology of *wi* is unknown, it could be MHG *wie* or *wâr*, see: https://www.woerterbuchnetz.de/ElsWB?lemid=W00044.

(5) *Er hätt gern siner Büch gfillt mit*
he would have like his stomach filled with
***de Schotte, wo** d' Söi gfresse hàn.*
the pods, that the pigs eaten have.
'He would have liked to fill his stomach with the pods that the pigs were eating' (T9, s8)

The invariant relative marker *wo / wu / wü / wi* is also used in cases where a pronominal adverb would be used to introduce a relative clause in German. In such sentences, both the relative marker and the pronominal adverb are annotated with the same dependency as shown in example (6), with the Standard German equivalent in green.

(6) 'The chair on which I sit.'


acl:relcl
obl:lmod
obl:lmod
De Stüel , wo-n-9 i druff sitz
The stool , which I thereon sit
Der Stuhl , worauf ich sitze
DET NOUN PUNCT PRON PRON ADV VERB

Pronominal adverbs Similarly to Standard German and other West-Germanic languages, Alsatian has pronominal adverbs. Whereas pronominal adverbs in Standard German can consist of different types of adverbs and prepositions (Pittner, 2024, p. 2–3), they can only contain adverbs starting with *dr-, de-* in Alsatian. They are often reinforced with a preposed *do* 'there' as in *do durich / dodurich*, written merged or split depending on the author. The pronominal character of these adverbs means that they can replace prepositional phrases, although there are some restrictions about the type of preposition phrases they can replace (see Pittner (2024) for more details in Standard German).

Pronominal adverbs can have different functions in Alsatian, and they can modify either a noun (7) or a verb (8–11). They can replace an element which occurs earlier in the sentence (anaphora, 8) or an element which occurs later in the sentence (cataphora, 9). They can also be used to refer to a physical element present in the context of the utterance (deixis, 10).

(7) *S Zìel **devùn** ìsch, d' Rachte ze vernìchte*
The goal **thereof** is, the rights to destroy
With `nmod:poss`. 'Its goal is to destroy the rights.' (T6, s82)

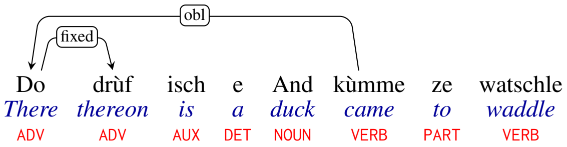
(8) *D' Eh derf nùmme gschlosse ware,*
The marriage may only concluded be,
wànn beidi Hochzitter fréi ùn vollstandi
if both spouses free and fully
***demìt** inverstànde sìnn.*
therewith agreed are.
'The marriage shall be entered into only with the free and full consent of the intending spouse.' (T6, s39)

(9) *ùn wil se sich **defir***
and because they themselves **therefor**
entschìdde hàn, de soziàle Fùrtschrìtt ze
decided have, the social progress to
férdere ùn besseri Lawersbedìngùnge mìt
promote and better life conditions with
ere greessere Fréiheit ze schàffe.
a bigger freedom to establish.
'And because they have decided to promote social progress and better standard of life in larger freedom.' (T6, s7)

(10) ***Do druf** kànnsch sìtze.*
There thereon can you sit.
Pointing to a chair: 'You can sit on this.'

We annotate pronominal adverbs with the POS `ADV` and the dependency relation `obl` if they modify a verb (8–11), and `nmod` if they modify a noun (7). We chose to use a different dependency than for other adverbs (usually `advmod`) in order to highlight their specificity. We chose `obl` and `nmod` since these dependencies would be used for the prepositional phrase which pronominal adverbs replace, for example compare *s Ziel devùn* and *s Ziel vùn de Tàte*. When the adverb *do* is used to reinforce a pronominal adverb, we annotate the second element as `fixed` of the first one, since it can also be written as one word:

(11) 'Just then a duck came waddling around.' (T8, s6)

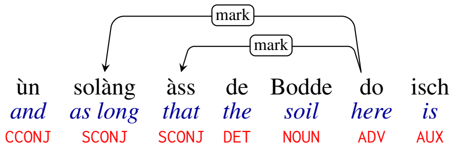

obl
fixed
Do drùf isch e And kùmme ze watschle
There thereon is a duck came to waddle
ADV ADV AUX DET NOUN VERB PART VERB

Double/Complex subjunctions In Alsatian, many subordinating conjunctions (subjunctions) that would be formed with one word in Standard German are formed with a preposition followed

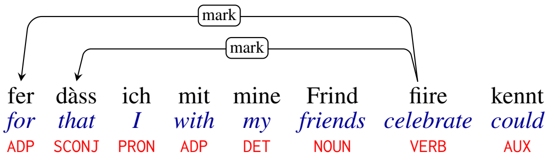
[9] In this example, the relative marker *wo* is followed by an epenthetic *n* to avoid hiatus with the following *i*, resulting in the form *wo-n-i* [ʋʊni] instead of *wo i* [ʋʊ.i]. See section 3.4 Epenthesis for more details.

by the subjunction ‘dàss/àss’[10] or ‘wie’, e.g. *fer dàss* (Ger. ‘damit’), *for/ebb dàss* (Ger. ‘bevor’), *sobàl wie* (Ger. ‘sobald’) (see Jung, 1983, p. 246). This is also found in Low Saxon (Siewert and Rueter, 2024, p. 15980) and Bavarian (Blaschke et al., 2024, p. 10926). This construction is also found with two subjunctions, for which an additional ‘dàss/àss’ would not have been needed, and possibly appeared by analogy with the construction preposition + subjunction (Huck et al., 1999, p. 57–60). For example, the following complex/double subjunctions are found: *obwohl àss, trotzdem àss, noochdem dàss, wurum dàss*. We annotate this construction as follows: the first element keeps its original POS, usually `ADP` (*fer, vor*), `ADV` (*werum*) or `SCONJ` (*obwohl, sowyt*), and the second element is always annotated with the POS `SCONJ`: *dàss / àss / wie*. Both are linked to the subroot with `mark`.

(12) ‘and as long as the soil is here [...]’ (T5, s180)



(13) ‘[...] so I could celebrate with my friends.’ (T9, s26)

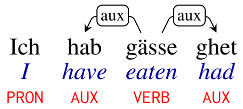


3.3.3 Verb Phrases

Lack of preterite One of the differences between Alsatian and Standard German is the total loss of the preterite tense, which led to new constructions to express it. The habitual aspect is instead expressed using *àls/àss*,[11] which we annotate `ADV` `obl:tmod`, as in (14). Because of the loss of the preterite tense, the past anterior is built using the past participles of both the verb and the auxiliary, both annotated using `aux` (see 15).

(14) *Ich bin* ***ass*** *in de Kinnes mit ’m.*
I am ≈often in the cinema with him.
‘I used to go to the cinema with him.’ (Jung, 1983, p. 190)

(15) ‘I had eaten.’



Lack of future tense The future tense is not grammatically differentiated in Alsatian and the present tense is used to speak about an action that will happen in the future. The verbal form which resembles Future I and II in Standard German (*werden* + infinitive; in Alsatian with *wĕrre / ware / warde*, see Jenny and Richert (1984, p. 37)) is instead used as a modal verb to indicate an hypothesis or an assumption. In the following example, an assumption is made since the speaker knows Peter’s habits:

(16) *Wü isch de Peter? — Är* ***wurd*** *widder*
Where is the Peter? — He **must** again
im Gaarte hucke.
in the garden hang out.
‘Where is Peter? — He must be hanging out in the garden.’

Periphrastic present with *düen* As in some other southern German non-standard varieties, *düen* (Ger. *tun*, Eng. *do*) can be used as an auxiliary in the present tense, although its use is different than in English. In Alsatian, *düen* stresses the active, dynamic nature or the effectiveness of the action, and can also express a prospective mood to carry out the action (Kleiber and Riegel, 2005).

(17) *Sie* ***düen*** *Füesball speele.*
They **do** football play.
Here: ‘They are playing football.’

Beneficiary voice Another specificity found in Alsatian is the beneficiary voice (Jenny and Richert (1984, p. 29), see under *middle voice*),[12] used with ditransitive verbs such as *give, steal*. The ‘beneficiary’ of the verb (in the dative case) becomes the subject, the auxiliary *bekumme/krieje* ‘to receive’ is conjugated in present tense, and the main verb is in its past participle form (see 18). We annotate these forms with the `:bfoc` subrelation (beneficiary focus, see UD Voice=Bfoc):

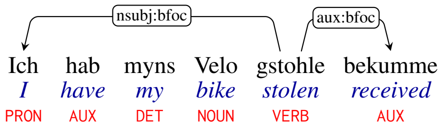
[10] The choice between ‘dàss’ or ‘àss’ depends on the variety, the speaker, stylistic choices, as well as the phonetic context.

[11] See meaning 1 in Martin and Lienhart (1899/1907)’s Alsatian dictionary: https://www.woerterbuchnetz.de/ElsWB?lemid=A00487.

[12] The term *middle voice* refers to a different phenomenon in UD guidelines, and we have thus decided to use UD’s *beneficiary-focus voice*, which corresponds better to this phenomenon: https://universaldependencies.org/u/feat/Voice.html#Bfoc.

nsubj:bfoc and aux:bfoc, similarly to the annotation of the passive voice.

(18) ‘I got my bike stolen.’

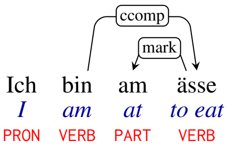

nsubj:bfoc
aux:bfoc
Ich hab myns Velo gstohle bekumme
I have my bike stolen received
PRON AUX DET NOUN VERB AUX

Conditional Mood In Alsatian, the German Konjunktiv II, which we call *conditional*, is not built with *werden* → *würde* as in Standard German, but with the auxiliary *düen* → *dät/dat* (19), or in central dialects with *gan* → *gat*[13] (20). The German Konjunktiv I is only used for the two auxiliaries *hàn* → *héig*, *sin* → *séig* in the varieties of Mulhouse and to the south of the city (Wikiversité, 2023).

(19) *Was* ***dätsch*** *dü mache?*
‘What would you do?’ (T8, s29)

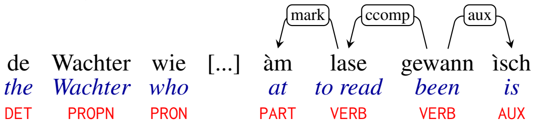
(20) *Wa dü enne g’kännt hätsch, d’rno*
If you him known would have, then
wär ’s andersch komme, unn d’rno
would be it different came, and then
gäht *’s besser met emm stehn.*
would it better with him stand.
‘If you had known him, things would have turned out differently and he would have been better off.’[14]

Progressive Aspect Similarly to other German non-standard varieties, Alsatian expresses the progressive aspect using the present and past tenses. It is built with the auxiliary *sin* ‘to be’ in the present tense, and a nominalised verb preposed by *am*. We chose to annotate it as a fully grammaticalised construction, thus treating the main semantic verb (e.g. *asse* ‘eat’) as VERB preposed by *am* PART, the whole group being a ccomp of the auxiliary *sin*, which is annotated as VERB. See (21) for an example in the present tense and (22) in the past tense.



(21) ‘I am eating.’

(22) ‘The [blacksmith] Wachter, who was reading [the poster with his apprentice].’ (T5, s12)


mark
ccomp
aux
de Wachter wie [...] àm lase gewann ìsch
the Wachter who at to read been is
DET PROPN PRON PART VERB VERB AUX

3.3.4 Other Domains

2SG dropped subject pronouns (null subjects) Similarly to Bavarian (Blaschke et al., 2024, p. 10,926) and other non-standard German varieties, second person subject pronouns can be omitted in some contexts in Alsatian (see Hoff (2024a), Hoff (2024b)). There were a few instances of this phenomenon in our corpus (see 23). Since we have not annotated morphological features, the verb in this construction is treated like any other verb, and the absence of a subject pronoun is not annotated for.

(23) ***Brüchsch*** *ken Àngscht ze hàn for mich, papa*
You need no fear to have for me, dad
‘You don’t need ot be afraid for me, dad.’ (T1, s1)

3.4 Use of Features

The only (non-miscellaneous) features used in the corpus were Foreign=Yes and Epenthesis=Yes. See Table 5 in the appendix for details of the statistical distribution of these features in the corpus.

Foreign The feature Foreign was used to indicate loanwords which were not adapted to Alsatian phonology and orthography. For example, the word *Mösjö* ‘mister’ is adapted, while *Monsieur* is considered a French loanword, not adapted to Alsatian. This feature is always accompanied by the use of the miscellaneous feature Lang, further indicating the language the loanword originates from. In our corpus, some of the foreign languages present were for example: fr (French), de (German), en (English), oci (Occitan), etc.

Proper nouns were treated differently depending on their type: personal names were not annotated as loanwords (ex: *Auguste*), while names of countries (*Bolivie*) or languages (*Creole*) were annotated as loanwords when a non-Alsatian form was used, based on its spelling. Acronyms were annotated as loanwords when they were made up of foreign elements or words. For example, *ONU* ‘**O**rganisation des **N**ations **U**nies’ (in English, *UN – United Nations*) was annotated as a French loanword.

[13] As described by Philipp et al. (1985, p. 5,846), the area extends approximately from Rosheim to the Munster valley. The exact localities can also be found on the map ‘täte’ in the Wenker (1889/1923) Atlas. See the dark blue backslashes, online at: https://apps.dsa.info/sprachgis/atlas/wa/538.

[14] From the Wenker (1889/1923) Atlas, see question 18 in the locality Bourgheim: https://apps.dsa.info/wenker/transliteration/18607.

Epenthesis The feature Epenthesis=Yes was used to indicate words to which an epenthetic consonant was added, for example: *So-**n**-e scheens Hardfir* (T5, s173) 'such a beautiful fire'. An epenthetic consonant, usually *n*, but also *w* in some varieties of Alsatian,[15] is added between two vowels at a word boundary (Wikiversité, 2023). This phenomenon, also called 'Binde-n', is typical of High Alemannic dialects and occurs irrespective of vowel quality (Ortmann, 1998). The type of element/host to which *n* is cliticised plays however a role in its distribution: for example, it never appears on non-finite verb forms (see Ortmann (1998) for more details about this phenomenon and a theoretical explanation). When annotating this phenomenon, we decided to merge the epenthetic consonant to the previous word, as was also done in the Swiss German UD corpus.

Gloss and Lemma The miscellaneous features Gloss[fr] and Lemma[de] were used to provide word-for-word translations in French and lemmas in Standard German. The gloss in French corresponds to an inflected/conjugated form, while the lemma in German always indicates a 'base' form in the nominative, non-inflected for gender, number, or tense (except for some specific determiners and pronouns, for which gender and number is more relevant). For some words, we indicated multiple German lemma: the first corresponds to the German word with the same etymon, and the second corresponds to modern use in Standard German. For example, *dummel di!* 'hurry!' is annotated Lemma[de]=tummeln/beeilen.

4 Conclusion

We have presented our corpus of Alemannic Alsatian dialects annotated following Universal Dependencies guidelines, which is the first one for this dialect. The corpus was pre-annotated using a variety of tools, and annotated by two annotators, while creating and further developing annotation guidelines for Alemannic Alsatian dialects. While many aspects of Alsatian grammar are similar to Standard German, a few specificities were identified and presented in this article. Some syntactic phenomena and annotation decisions for Alsatian were presented and compared to the existing literature and resources on UD corpora for Bavarian, Low Saxon, and Luxembourgish, German varieties related to Alsatian. The corpus described in this paper is undergoing its final review process for addition to the UD repository. It will be available from the following repository: https://github.com/UniversalDependencies/UD_Alemannic-DIVITAL and will add to the resources on non-standard German varieties.

Limitations

Translations Since some of the source texts for our corpus were translated from French, we cannot determine the extent to which the translations were influenced by French and/or Standard German, and to which extent this data differs from naturally occurring Alsatian data. Similarly, annotation decisions were heavily influenced by annotation guidelines for Standard German, which were more accessible and more detailed than guidelines for other non-standard German varieties.

Representation of Alsatian dialects Our project focuses on Low Alemannic dialects and is thus not representative of the whole region: High Alemannic is only represented by a few sentences and Franconian varieties are absent. Furthermore, a great majority of the texts in our corpus are in Northern Low Alemannic. Oral genres and transcriptions are also underrepresented in the corpus, in comparison to written genres.

Ethical considerations

The data used for this project is either freely available from accessible sources, available for research purposes,[16] or the permission to use them for this project was granted by the authors or translators (applies to texts T9 and T11). Excerpts were used in Phases 1 and 3, in accordance with the right to quote. The translators (for texts T5-7-8-11) and the annotators involved in this project were fully compensated for their contributions.

Acknowledgments

This work has been carried out within the framework of the ANR-21-CE27-0004 DIVITAL project supported by the French National Research Agency. We would like to thank the translators organizations and authors who contributed to the

[15] For more details, see Sakumoto (2024).

[16] T10 is licensed under CC-BY-NC-SA and, for the translation of the Universal Declaration of Human Rights (T6), see https://www.ohchr.org/en/human-rights/universal-declaration/universal-declaration-human-rights/about-universal-declaration-human-rights-translation-project

corpus contents: Yves Bisch, Conseil départemental du Haut-Rhin, OLCA, Adrien Fernique, Carole Werner and Michel Feltin-Palas.

References

Noëmi Aepli and Simon Clematide. 2018. Parsing Approaches for Swiss German. In *Proceedings of the 3rd Swiss Text Analytics Conference (SwissText), Winterthur, Switzerland*.

Nathanaël Beiner. 2022. *Quelle(s) norme(s) pour l'écriture de l'alsacien en 2022 ?* Master's thesis, Université de Strasbourg.

Delphine Bernhard, Nathanaël Beiner, and Barbara Hoff. 2025. Pre-annotation Matters: A Comparative Study on POS and Dependency Annotation for an Alsatian Dialect. In *Proceedings of the 19th Linguistic Annotation Workshop*. To appear.

Ernest Beyer. 1964. *La Palatalisation vocalique spontanée de l'alsacien et du badois : sa position dans l'évolution dialectale du germanique continental*. Thèse d'État, Université de Strasbourg.

Verena Blaschke, Barbara Kovačić, Siyao Peng, Hinrich Schütze, and Barbara Plank. 2024. MaiBaam: A multi-dialectal Bavarian Universal Dependency treebank. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10921–10938, Torino, Italia. ELRA and ICCL.

Emanuel Borges Völker, Maximilian Wendt, Felix Hennig, and Arne Köhn. 2019. HDT-UD: A very large Universal Dependencies treebank for German. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 46–57, Paris, France. Association for Computational Linguistics.

Philippe Boula de Mareüil, Frédéric Vernier, and Albert Rilliard. 2018. A Speaking Atlas of the Regional Languages of France. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, page 6, Miyazaki, Japan.

Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. 2020. When collaborative treebank curation meets graph grammars. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5293–5302, Marseille, France. European Language Resources Association.

Bruno Guillaume. 2021. Graph matching and graph rewriting: GREW tools for corpus exploration, maintenance and conversion. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–175, Online. Association for Computational Linguistics.

Barbara Hoff. 2024a. L'omission du sujet référentiel en alsacien, comparée à d'autres variétés germaniques. *Cahiers du plurilinguisme européen*, 16.

Barbara Hoff. 2024b. Reddsch Elsassisch? Referential null subjects in Alsatian, compared to other Germanic varieties. Master's thesis, University of Oslo.

Dominique Huck, Arlette Laugel, and Maurice Laugner. 1999. *L'élève dialectophone en Alsace et ses langues : l'enseignement de l'allemand aux enfants dialectophones à l'école primaire*. Oberlin.

Alphonse Jenny and Doris Richert. 1984. *Précis pratique de grammaire alsacienne: en référence principalement au parler de Strasbourg*. ISTRA.

Edmond Jung. 1983. *Grammaire de l'alsacien, dialecte de Strasbourg avec indications historiques*. Oberlin.

Georges Kleiber and Martin Riegel. 2005. *Les périphrases düen + verbe à l'infinitif en alsacien: Un auxiliaire modal à tout faire*, pages 171–184. John Benjamins Publishing Company.

John MacFarlane, Albert Krewinkel, and Jesse Rosenthal. 2025. Pandoc.

Ernst Martin and Hans Lienhart. 1899/1907. *Wörterbuch der elsässischen Mundarten*. Karl J. Trübner.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.

Albert Ortmann. 1998. *Consonant epenthesis: its distribution and phonological specification*, pages 51–76. Max Niemeyer Verlag, Berlin, Boston.

Marthe Philipp, Arlette Bothorel-Witz, and Jean-Jacques Brunner. 1985. Parlers alsaciens. In *Encyclopédie de l'Alsace*, otfried-rhin edition, volume 10, pages 5838–5853. Publitotal.

Karin Pittner. 2024. Pronominal Adverbs in German: A Grammaticalization Account. *Journal of Germanic linguistics*, 36(1):1–31.

Alistair Plum, Caroline Döhmer, Emilia Milano, Anne-Marie Lutgen, and Christoph Purschke. 2024. LuxBank: The first Universal Dependency treebank for Luxembourgish. In *Proceedings of the 22nd Workshop on Treebanks and Linguistic Theories (TLT 2024)*, pages 30–39, Hamburg,Germany. Association for Computational Linguistics.

Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. 2017. Udapi: Universal API for Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 96–101, Gothenburg, Sweden. Association for Computational Linguistics.

Daisuke Sakumoto. 2024. The insertion of voiced labiodental fricative to avoid hiatus in Alsatian: phonological and morphosyntactical conditions of occurence, and diachronical formation process. *Studies on Enunciative Linguistics*, 3:126–153.

Alessio Salomoni. 2017. Toward a Treebank Collecting German Aesthetic Writings of the Late 18th Century. In *Proceedings of the Fourth Italian Conference on Computational Linguistics CLiC-It 2017*, pages 192–197, Torino, Italy. Accademia University Press.

Janine Siewert and Jack Rueter. 2024. The Low Saxon LSDC dataset at Universal Dependencies. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15976–15981, Torino, Italia. ELRA and ICCL.

Daniel Steiner and Raymond Matzen. 2016. *D'Biwel uf Elsässisch*. Éditions du Signe, Strasbourg.

Dejan Stosic, Saša Marjanović, Delphine Bernhard, Xavier Bach, Myriam Bras, Laurent Kevers, Stella Retali-Medori, Marianne Vergez-Couret, and Carole Werner. 2024. The ParCoLab parallel corpus and its extension to four regional languages of France. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16014–16023, Torino, Italia. ELRA and ICCL.

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. Association for Computational Linguistics.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Georg Wenker. 1889/1923. *Sprachatlas Des Deutschen Reichs.*

Wikiversité. 2023. Alsacien/grammaire/annexe/synthèse complète — wikiversité. `https://fr.wikiversity.org/wiki/Alsacien/Grammaire/Annexe/Synth%C3%A8se_compl%C3%A8te`. [Online; page available August 22, 2023; accessed February 28, 2025].

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Arofat Akhundjanova, Furkan Akkurt, Gabrielė Aleksandravičiūtė, Ika Alfina, Avner Algom, Khalid Alnajjar, Chiara Alzetta, Erik Andersen, Matthew Andrews, and 633 others. 2024. Universal dependencies 2.15. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, and 43 others. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

A Appendix

Relation	Frequency	Relation	Frequency	Relation	Frequency
punct	3,188 – 16%	nmod	207 – 1%	nmod:lmod	65 – 0%
det	2,207 – 11%	advcl	186 – 0%	det:predet	55 – 0%
case	1,526 – 7%	xcomp	177 – 0%	expl:pv	54 – 0%
nsubj	1,372 – 6%	discourse	176 – 0%	reparandum	44 – 0%
root	975 – 4%	parataxis	168 – 0%	acl	41 – 0%
conj	974 – 4%	appos	153 – 0%	flat:name	31 – 0%
aux	890 – 4%	ccomp	149 – 0%	obl:agent	24 – 0%
advmod	813 – 4%	advmod:emph	141 – 0%	nmod:tmod	23 – 0%
obj	785 – 3%	obl:tmod	138 – 0%	dislocated	23 – 0%
cc	782 – 3%	expl	135 – 0%	csubj	15 – 0%
obl	596 – 3%	aux:pass	128 – 0%	orphan	9 – 0%
amod	572 – 2%	compound:prt	125 – 0%	compound	5 – 0%
mark	542 – 2%	nummod	115 – 0%	cc:preconj	5 – 0%
cop	289 – 1%	obl:arg	111 – 0%	nsubj:outer	2 – 0%
advmod:tmod	286 – 1%	nsubj:pass	108 – 0%	csubj:outer	1 – 0%
obl:lmod	265 – 1%	fixed	104 – 0%	advcl:relcl	1 – 0%
nmod:poss	243 – 1%	flat	103 – 0%	obl:poss	1 – 0%
acl:relcl	212 – 1%	advmod:lmod	81 – 0%	det:preconj	1 – 0%
det:poss	208 – 1%	vocative	66 – 0%		

Table 3: Statistics for dependency relations. For each relation, both the absolute and relative frequency are indicated.

Part-of-Speech Tag	Frequency	5 most frequent tokens
PUNCT – Punctuation	3,188 – 16%	, . ! : . . .
NOUN – Noun	2,901 – 14%	*Racht, Mensch, Sproch, Herr, Rachte*
DET – Determiner	2,705 – 13%	*de, m, e, d’, s*
VERB – Verb	1,795 – 9%	*het, hàn, gànge, redde, gemàcht*
ADP – Adposition	1,638 – 8%	*vùn, in, ìn, fer, i*
PRON – Pronoun	1,455 – 7%	*wie, mr, mer, se, ’s*
ADV – Adverb	1,358 – 6%	*do, so, noch, no, àls*
AUX – Auxiliary	1,348 – 6%	*het, isch, ìsch, hàn, esch*
ADJ – Adjective	1,098 – 5%	*gànz, besser, kleine, fréi, scheen*
CCONJ – Coordinating conjunction	727 – 3%	*ùn, un, odder, oder, Un*
PROPN – Proper noun	413 – 2%	*Peter, Frànkrich, Hamel, Elsàss, Kobüs*
SCONJ – Subordinating conjunction	314 – 1%	*àss, wie, wenn, wànn, dàss*
PART – Particle	311 – 1%	*ze, nit, nìt, net, nitt*
NUM – Numeral	222 – 1%	*zwei, drei, 2, sechs, drissig*
INTJ – Interjection	169 – 0%	*Ja, ja, hein, euh, Jo*
X – Other	43 – 0%	*ta, bon, BA, BE, BI*
SYM – Symbol	11 – 0%	%, ∗, †, &, *n*°

Table 4: Statistics for POS tags. For each tag, both the absolute and relative frequency are indicated, as well as the five most frequent tokens

Feature	Frequency
Foreign=Yes	490 – 2%
Epenthesis=Yes	35 – 0%

Table 5: Statistics over features in the corpus. Both the absolute and relative frequency are indicated.

Expanding the Universal Dependencies Ancient Hebrew Treebank with Constituency Data

Daniel G. Swanson
Department of Linguistics,
Indiana University,
dangswan@iu.edu

Abstract

This paper presents an effort to expand the annotation pipeline for the Ancient Hebrew Universal Dependencies treebank to make use of additional data, resulting in the addition of over 4000 sentences and roughly 100K words to the released treebank. The resulting treebank contains 5500 sentences and 145K words and the incorporation of converted constituency data has resulted in an annotation process which requires manual intervention in only around 15-20% of sentences, even in previously unseen genres.

1 Introduction

Swanson and Tyers (2022) developed a rule-based parser and used it to produce a UD treebank of portions of the Hebrew Scriptures. In this paper, we extend their processing pipeline to addtionally take input from a partial constituency treebank.

The Hebrew Scriptures are a collection of 39 books primarily written in the first millennium BC in Ancient Hebrew (with a few passages in Aramaic) which were arranged and codified in their current form over the course of the first millennium AD. They are also known as the Tanakh, an acronym of the Hebrew names of the 3 main divisions: תורה /torah/ "law"[1], נביאים /nevi'im/ "prophets" (a category which also includes several books of narrative history), and כתובים /ketuvim/ "writings".

The Universal Dependencies (UD) project (Nivre et al., 2020) is a collaborative effort to create a collection of treebanks in a single cross-linguistically consistent annotation scheme so as to better facilitate studying syntax in multiple languages.

This paper is organized as follows: Section 2 describes the existing corpora used to create the treebank, Section 3 explains the existing pipeline and our modifications and evaluation, Section 4 discusses changes we made to the annotation guidelines for Ancient Hebrew, Section 5 provides statistics on the resulting treebank, and Section 6 concludes.

2 Data Sources

The data for this project comes from two sources: The Biblia Hebraica Stuttgartensia Amstelodamensis (BHSA) and the MACULA project, both of which annotate the same underlying text. Each corpus contributes valuable but incomplete data to the task of producing a UD treebank.

The BHSA (Peursen et al., 2015) provides extensive morphological annotation of the text, and some light semantic annotation. Its syntactic annotations, however, are extremely limited, with the structure of most sentences being restricted to a two-layer constituency tree (phrases and clauses), as shown in Figure 1. Deriving slightly more detailed trees from the BHSA data is possible[2], but Swanson and Tyers (2022) nonetheless ended up building a system that was much closer to a parser than to a treebank converter.

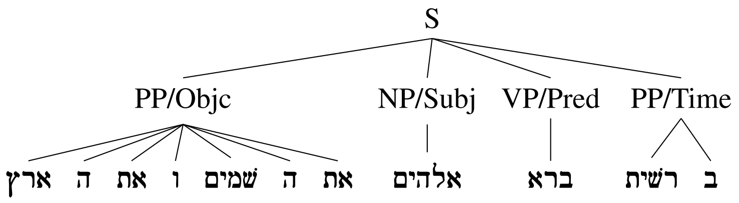
In contrast, MACULA (Clear Bible, 2022), which was released after Swanson and Tyers (2022), has full syntactic information up to the clause level, as shown in Figure 2. However, for more complex sentences, it often leaves the upper layers underspecified, such that each clause is fully annotated, but the relations between clauses are not, as shown in Figure 3.

Fortunately, some of BHSA's more semantic features can help fill this gap. One of the most important is a feature called `txt`, which marks the "level" of the text, in particular distinguishing between narrative, quotations, and quotations embedded within quotations, which is usually sufficient to resolve

[1] Instances of Hebrew script in this paper are followed by a transliteration in slashes according to the ALA-LC scheme (Barry, 1997) and an English translation in quotes.

[2] See, for example https://github.com/ETCBC/trees.

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 23–31
August 28-29, 2025 ©2025 Association for Computational Linguistics



ברשית ברא אלהים את השמים ואת הארץ

ארץ	ה	את	ו	שמים	ה	את	אלהים	ברא	רשית	ב
’arets	ha	’et	ve	shamayim	ha	’et	’elohiym	bara’	reshiyt	be
land	the	ACC	and	sky	the	ACC	God	created	beginning	in
subs	art	prep	conj	subs	art	prep	subs	verb	subs	prep

“In the beginning, God created the sky and the land.”

Figure 1: The syntax tree and gloss of Genesis 1:1, as given by the BHSA (the source used for prior work). Phrase nodes are marked with both their category and their function label (here, “object”, “subject”, “predicate”, and “time”).

the attachment of clauses which are complements of speaking verbs or are simply coordinated. (Other types of subordination present further challenges, which are discussed below.)

3 Methodology

Swanson and Tyers (2022) used Constraint Grammar (CG) (Bick and Didriksen, 2015) as the basis for the following pipeline:

1. Convert BHSA to CG format
2. Parse with CG
3. Convert to CoNLL-U format
4. Apply UDapy to attach punctuation (Popel et al., 2017)
5. Manually review trees

For our work, we extend the pipeline to the process depicted in Figure 4 by adding the following steps:

Rule extraction This script converts the constituency structure of MACULA into a dependency structure. Each node which has multiple children (apart from the top-level sentence node) has attributes which specify which child is the head and what parsing rule generated the node. For each such node, the head is set as the parent of all the other children, and the children receive the rule name as a tag. This process is depicted in Figure 6. This step is substantially simpler than what often appears in other conversion projects, since it only requires a tree traversal to collect all the relations without needing a set of heuristics to identify the heads at each level (Arnardóttir et al., 2020; Chun et al., 2018; Kuzgun et al., 2021).

Alignment and arc projection This step determines the correspondences between word IDs in BHSA and MACULA. Since the two corpora represent the same underlying text, this is trivial at the sentence level, but presents some challenges at the word level due to differences in tokenization, some of which are shown in (1).

(1) לָבוֹא אֶל יִצְחָק אַבִיו אַרְצָה כְּנָעַן

/labo’ ’el yitsḥaḳ ’aviṿ ’artsah ken‘an/

BHSA	כְּנָעַן		אַרְצָה
MACULA	כְּנָעַן	ה-	אַרְצָ
UD	כְּנָעַן		אַרְצָה
Gloss	Canaan	LOC	the land of

	אָבִיו	יִצְחָק	אֶל	בּוֹא	לָ-
ו-	אָבִי	יִצְחָק	אֶל	בּוֹא	לָ-
ו-	אָבִי	יִצְחָק	אֶל	בּוֹא	לָ-
his	father	Isaac	toward	come	to

“in order to come to Isaac his father in the land of Canaan” (from Genesis 31:18)

Here we see the two most frequent divergences: MACULA treats the locative suffix as a separate unit, unlike BHSA and UD, and

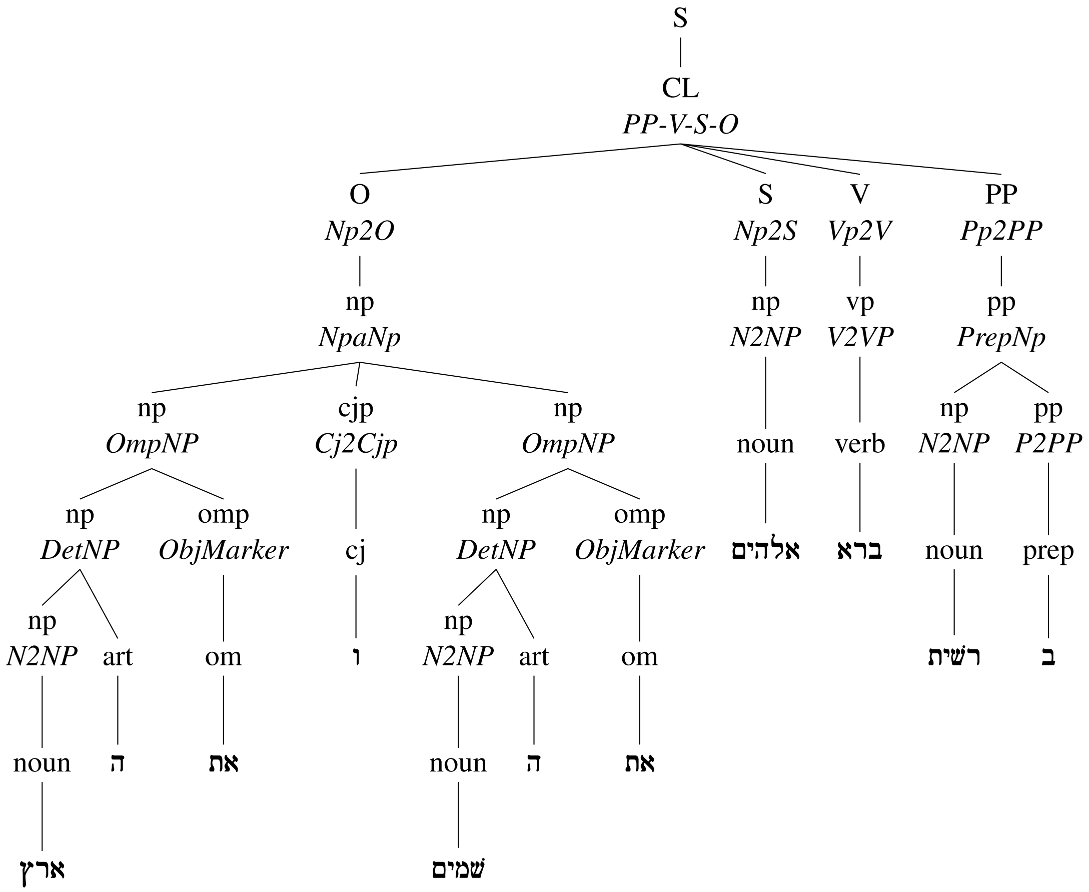


Figure 2: The syntax tree of Genesis 1:1, as given by MACULA (the source added in this work). Labels in *italics* are the names of the parsing rules that generated the nodes. The corresponding gloss can be found in Figure 1.

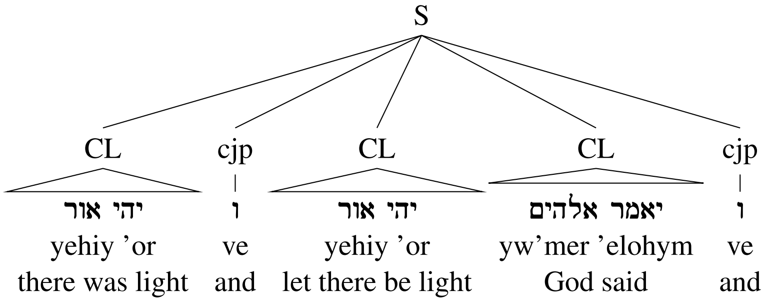


Figure 3: The top layer of the syntax tree of Genesis 1:3 in MACULA. While each `CL` (clause) node shown has full internal structure, none of the annotations give any indication of how these constituents are related to one another, hence the continued need for the data from BHSA.

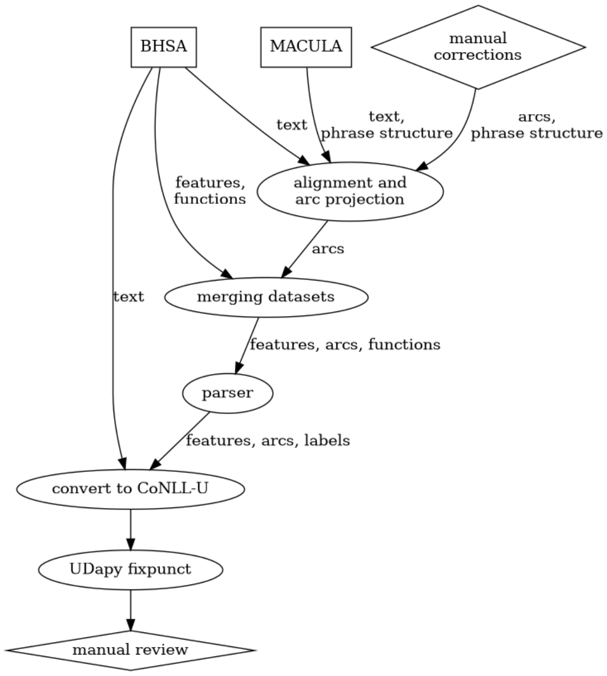


Figure 4: The relationship between the different soruces of data and the scripts that combine them. Rectangular nodes represent external corpora, oval nodes represent scripts and tools, and diamond nodes represent data reviewed or created by the authors. Edge labels indicate the information that is passed between the nodes.

BHSA does not treat pronominal suffixes as separate units, unlike MACULA and UD. The alignment script applies a set of rules describing these divergences to the two texts until it achieves an exact match. If such a match is not found, it reports the sentence to the developer for review.

Having generated these alignments, the script converts the constituency structure of the MACULA nodes into dependency arcs between BHSA words. To do this, it takes the head of each phrase node (which MACULA specifies) and adds an arc from it to the head of each of its siblings, using the rule name as the arc label. Each word is then identified with all the phrases it is the head of, producing a dependency tree between words. This process is depicted in Figure 6.

Manual corrections This consists of two sets of files, both of which override specific parts of the Alignment and arc projection step. One set identifies nodes in the MACULA data, and changes which child is marked as the head. One case where this occurs is in copular clauses where the heuristics in the rule extraction script are not always able to select the correct predicate, such as when the correct predicate is a locative adverb. The second set identifies a BHSA word, and changes which word is its head and/or adds tags (including the dependency label). This is most frequently used in cases where the parser fails to correctly attach subordinate clauses. Examples of both types are given in Figure 5. These two sets of overrides replace the previous system, in which the full CoNLL-U of any tree that required manual correction would be copied to a separate file.

```
#60
w1245 w1251
w1247 w1251 @obl
#77
w1607 _ @conj

#393
0101701400120180 2
```

Figure 5: Examples of entries in the manual override system for Genesis. The first two entries specify that particular words in sentence 60 should have different heads, and the second additionally specifies that the relation label should be set to obl. The third, meanwhile, specifies that a particular word in sentence 77 should have the relation conj, but should keep whatever head was extracted from MACULA. Finally, the last entry specifies that in sentence 393, when extracting rules from a particular MACULA node, daughter node 2 should be treated as the head.

With these changes, we were able to replace significant portions of the original parser with a set of rules that largely amount to a decision tree converting MACULA's rule names into UD relations, using BHSA morphological and semantic labels to disambiguate them (such rules now make up roughly one third of all rules in the parser). An example is given in (2).

(2)

```
WITH NOMAPPED (NpAdjp) {
  MAP @det (prde) OR (ppde) ;
  MAP @acl:relcl (verb) ;
  MAP @nummod (adjv ordn) ;
  MAP @amod (*) ;
} ;
```

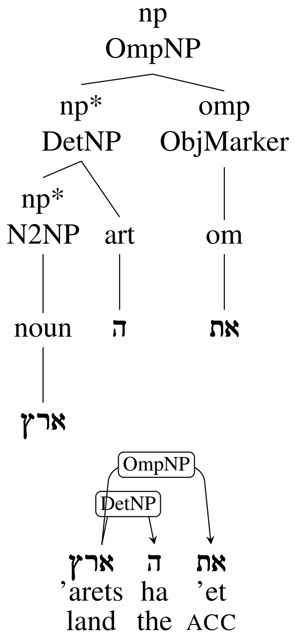
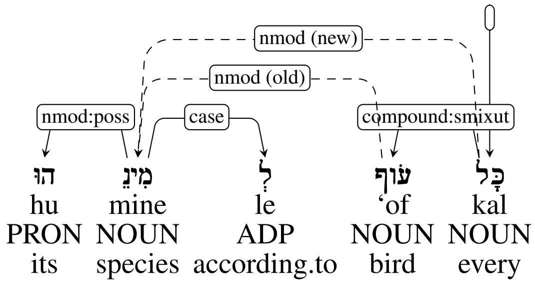


Figure 6: The process of converting from MACULA trees into initial dependencies. Nodes marked with * are the heads of their parent rules. Note that the rule of the N2NP node does not appear in the resulting tree, because it has only one child. These labels will be changed case and det by the Constraint Grammar rules.

This fragment uses the recently added compound rules (Swanson et al., 2023) to create a nested conditional, where the first line restricts the subsequent rules to apply only to words whose rule label is NpAdjp (adjective phrase modifying a noun phrase). The rules check first if the word is tagged as demonstrative, then if it is a verb, then if it is an ordinal number, and finally apply amod (adjectival modifier) if none of the other conditions apply.

As we adjusted the parser to use MACULA input, we regularly checked its output against the previously validated trees (both those that appeared in the released treebank and another roughly 300 trees which had not been released since they did not constitute a complete document). In the process, we discovered and fixed a number of inconsistencies, largely in modifier attachment, such as the one shown in (3), where the dashed lines indicate the old analysis and the new analysis.

(3)



Quantifiers in Hebrew are morphologically nouns, which combine with other nouns via a highly productive compounding construction. In the existing trees, it was common for modifiers on such phrases, such as the phrase meaning "according to its species" in (3), to attach to the lexical noun (here "bird") which would in such cases arguably be the semantic head, rather than the quantifier which is the morphosyntactic head. We have updated these instances to be more consistent in their treatment of such constructions, so that modifiers are attached to the entire compound phrase unless there is a good reason to do otherwise (adjectives are still sometimes attached to the dependent noun if their agreement features do not match those of the head noun).

Once this updating was complete, we performed an analysis of the accuracy of the parser on unvalidated data. In virtually all orderings of the books of the Hebrew Scriptures, the book of Exodus is the second book after Genesis, and thus seemed a natural continuation of the project. According to the method of splitting sentences which had been implemented in the treebank, Exodus contains 1151 trees[3], 118 of which had already been validated. We then manually inspected the remaining 1033 trees, and judged 810 (78.4%) of them to be fully correct without further modification.

For the remaining 223 trees, most only needed corrections for a handful of words. In a number of cases, this was due to there being rule names which had not come up in the Genesis data which needed to be added to the section of the parser that converts rule names to dependency labels. The largest source of such instances was for sequences of coordinated phrases, because MACULA has a separate rule name for each possible sequence, such as NPNPaNPaNPaNP, indicating a sequence of five noun phrases with a conjunction between each pair

[3] This is slightly lower than the traditional number of verses because of a handful of instances where a long list of objects crosses verse boundaries, resulting in a verse which is not a sentence.

except the first, and NPNPaNPNPaNP which is similar but also lacks a conjunction between the third phrase and the fourth. (We later adjusted the rules which handle coordination to recognize the pattern of such rules rather than requiring a fixed list, which should reduce the number of unknown rules in future expansions.)

Interestingly, the parser had roughly 80% sentence-level accuracy almost regardless of the genre of the text, as shown in Table 1. There is a very slight drop in performance on narrative in comparison to other genres, though this is likely a result of the longer sentences. Genesis and Ruth, the texts the parser was developed for, are both almost entirely narrative. Exodus, meanwhile, also contains some songs, a legal code, and building instructions. The songs, despite being poetry and in a noticeably more archaic style than the surrounding narrative, were actually the sections where the parser performed best. The only trees that were marked as incorrect were two which contained a word that had not been included in the list of subordinating conjunctions and thus did not receive a part of speech tag. This high performance is probably due to the fact that the poetic sections contain relatively few subordinate clauses and feature slightly shorter sentences. Their divergence from other genres is partly lexical, which affects hardly any rules, and partly word order, which also has little effect in this case, because most of the relations that differ are for nominal arguments (subject, object, etc.) which are usually accepted from MACULA as-is.

After completing this analysis for Exodus, we applied a similar methodology to the subsequent books. For each one, we made a first pass through the book, marking the trees that were already correct and leaving the rest for further processing. Then, in the course of correcting the remaining trees, we made various improvements to the rules before starting on the next book. The results of this process are given in Table 2, and they show a general upward trend reaching 84% by the end of the project.

4 Annotation Decisions

In the process of revising the treebank, there were hundreds of local changes, such as the modifier attachment discussed above, and a handful of larger systematic ones. In this section, we present three of the latter kind: the introduction of fixed expressions, a change in the tokenization guidelines relating to quotations, and the use of the expletive pronoun relation.

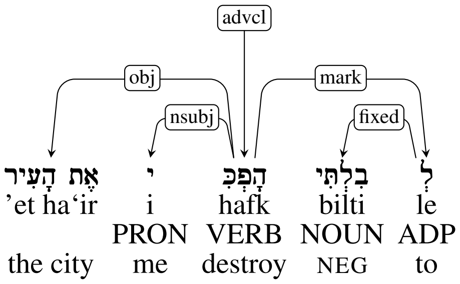
Further details about the annotation decisions of the treebank in general can be found in the UD documentation for Ancient Hebrew at https://universaldependencies.org/hbo/ and particularly the documentation of syntactic relations at https://universaldependencies.org/hbo/dep/.

4.1 Fixed Expressions

In the original version of the treebank, the fixed relation was only used for two constructions: עַד כִּי /‘ad ki/ and עַד אִם /‘ad ’im/, both of which function as subordinating conjunctions with the general meaning “until”. In both cases the phrase is composed of the preposition עַד /‘ad/ “until” followed by a subordinating conjunction. The preposition was previously tagged SCONJ in order to comply with the requirements of the validator. However, since UD version 2.16, the validator accepts the feature ExtPos as an alternative to having a particular part of speech tag for various checks, so עַד is now marked as ADP and ExtPos=SCONJ in this construction.

We have also identified a few more fixed expressions. One of these is a combination of כִּי /ki/ “because” and אִם /’im/ “if” to form כִּי אִם /ki ’im/ “unless”. This combination is both relatively frequent and also non-compositional, leading to our determination that fixed is an appropriate relation. There are a few cases where this sequence has compositional meaning, but in those cases the two conjunctions attach to different clauses, with כִּי /ki/ introducing a subordinate clause and אִם introducing a conditional which is further subordinate to that.

In addition, there is the לְבִלְתִּי /levilti/ “in order not to”, which is morphologically the preposition לְ /le/ “to” followed by the noun בִּלְתִּי /bilti/, which does not appear independently. A typical construction is that in (4).



(4)

Genre	Chapters	Sentences	Approved	Accuracy	Avg. Length
instruction	14	409	326	79.7%	25.9
narrative	20	479	371	77.5%	31.5
narrative and geneaology	2	32	20	62.5%	34.5
narrative and instruction	3	89	71	79.8%	28.6
poetry	1	24	22	91.7%	23.3

Table 1: Distribution of chapters and sentences in Exodus by genre. “Sentences” is the number of sentences examined and “Approved” is the number which did not require corrections after the initial MACULA conversion. “Accuracy” is the percentage of the total sentences that were approved in that initial pass and “Avg. Length” is the mean number of syntactic words in each sentence.

Book	Total	Prior	Remaining	First Pass	Accuracy
Exodus	1151	118	1033	810	78.4%
Leviticus	820	53	767	635	82.8%
Numbers	1179	116	1063	877	82.5%
Deuteronomy	879	21	858	722	84.1%

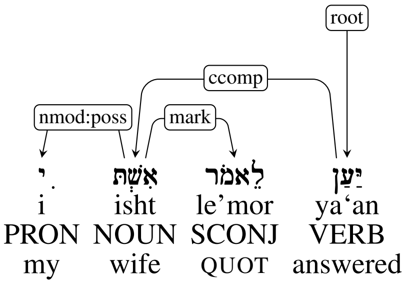
Table 2: The improvement of the parser over the course of the project. “Total” is the number of sentences in the book, “Prior” is the number of sentences validated in the course of Swanson and Tyers (2022), “Remaining” is the number of sentences that needed to be examined in the present work, “First Pass” is the number of sentences validated without adjustmnet, and “Accuracy” is the sentence-level accuracy of the parser on that book before making further updates to the rules.

“in order for me not to destroy the city”

Here the phrase לְבִלְתִּי /levilti/ precedes an infinitive verb, producing a negative purpose clause. We used `fixed` in this case since the phrase introduces a particular kind of clause and the noun which would be the head if this were a normal prepositional phrase is not used in any other context.

4.2 Quotations

Direct quotations in the text are often preceded by לֵאמֹר /le’mor/, which was originally tokenized as a single word and tagged as a subordinating conjunction, such as in (5).

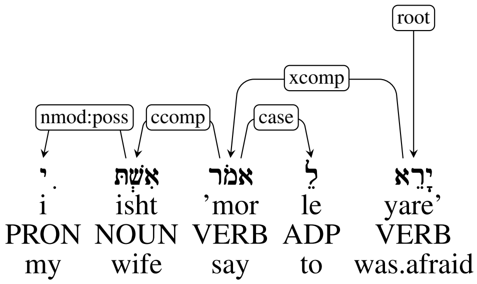


(5)

“He answered, saying ‘[she is] my wife’.”

However, in terms of morphology, this token is the preposition לְ /le/ “to” followed by the infinitive verb אֱמֹר /’emor/ “say” (infinitive verbs usually have prepositional prefixes in Hebrew, and ל is the most common one).

In fact, there are a few cases where the same surface string is a full verb, such as in (6).

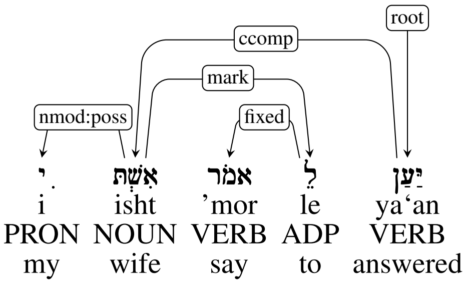


(6)

“He was afraid to say ‘[she is] my wife’.”

Here לֵאמֹר is a normal infinitive acting as a complement to the control verb יָרֵא /yare’/ “he was afraid”.

The result is that the tokenization guidelines called for prepositional prefixes such as ל to be split from their host words in all cases except the quotation marker. In light of the changes to the `fixed` and `ExtPos` guidelines discussed in the previous sections, we decided to remove the inconsistency here by tokenizing לֵאמֹר the same way everywhere and then marking the two pieces as a fixed expression. The analysis of (5) thus changes to that of (7).



(7)

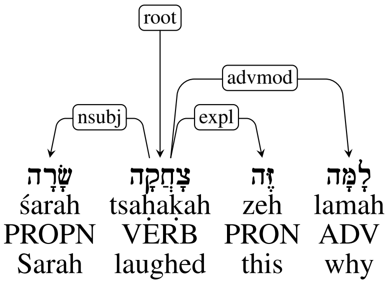
“He answered, saying ‘[she is] my wife’.”

And לֵ /le/ “to” has the feature `ExtPos=SCONJ` to satisfy the validator constraint that children of `mark` relations must be subordinating conjunctions.

4.3 Expletive Pronouns

The `expl` relation is used for nominals which fill a slot in the syntactic argument structure of a clause without filling any slot in the semantic argument structure. In the original version of the treebank, this relation was unused, but we found a few cases in which we concluded it was appropriate.

In questions, an interrogative pronoun or adverb will sometimes be followed by a demonstrative pronoun for emphasis, such as in (8).



(8)

“Why did Sarah laugh?” (from Genesis 18:13)

Here the sentence would be semantically identical if זֶה /zeh/ “this” were not present. Strictly speaking, it thus does not fill one of the main argument slots of the verb, and thus `expl` is not a perfect fit, but we determined that this was likely the best option for a nominal with no semantic role.

5 Treebank Statistics

This project has roughly quadrupled the total size of the Ancient Hebrew treebank. The exact numbers are given in Table 3.

Half of the added data (Exodus and Leviticus) was included in the 2.16 release of Universal Dependencies. The other half (Numbers and Deuteronomy) will be included in 2.17.

Book	Sentences	Tokens	Words
Genesis	1,494	25,282	36,822
Exodus	1,151	20,612	29,882
Leviticus	820	14,844	21,769
Numbers	1,179	20,221	28,925
Deuteronomy	879	17,421	26,171
Ruth	85	1,564	2,297
Total	5,608	99,944	145,866

Table 3: The sizes of the texts included in the treebank. Genesis and Ruth were previously released and the rest are new.

Book	Phrases		Arcs	
Genesis	11	(11)	164	(93)
Exodus	21	(19)	191	(108)
Leviticus	2	(2)	128	(67)
Numbers	2	(2)	128	(67)
Deuteronomy	7	(7)	161	(90)
Ruth	2	(1)	12	(8)
Total	45	(42)	784	(433)

Table 4: The number of manual overrides in each book. “Phrases” is the number of overrides to the headedness of MACULA nodes and “Arcs” is the number of overrides to heads or labels in the initial dependency structure. Numbers in parentheses indicate the number of distinct sentences.

Table 4 gives the frequency of manual overrides to the parser, which occur in around 8% of all sentences. This suggests 92% as a rough upper bound on the accuracy of the parser when applied to new data.

We also evaluated how much the new process changed the data that had already been released. To do this, we took the most recent released version of Genesis and Ruth (UDv2.15) and calculated the labeled and unlabeled attachment scores (UAS and LAS) between that version and our version. In order to make them properly comparable, we used UDapi to undo the tokenization change discussed in Section 4.2. The result was a UAS of 96.51 and an LAS of 95.39, which is consistent with our experience of a limited revision that nonetheless affected a substantial portion of the sentences in the corpus.

6 Conclusion

In this paper we have presented an effort to expand the Universal Dependencies Ancient Hebrew treebank by converting an existing partial constituency treebank. This process revealed various inconsistencies and areas for improvement in the existing annotations, which have now been fixed. In addition, it has greatly reduced the amount of manual effort required to produce new trees, since the accuracy of the parser is now high enough that a typical tree can be simply validated rather than leading to further debugging of the parser.

The treebank now includes approximately a quarter of the source text, and we intend to apply this process to annotate the remainder.

Acknowledgments

Thank you to Bryce Bussert and Amir Zeldes for fruitful discussions of the annotation decisions and to Sandra Kübler and Francis Tyers for reading drafts of this paper.

References

Þórunn Arnardóttir, Hinrik Hafsteinsson, Einar Freyr Sigurðsson, Kristín Bjarnadóttir, Anton Karl Ingason, Hildur Jónsdóttir, and Steinþór Steingrímsson. 2020. A Universal Dependencies conversion pipeline for a Penn-format constituency treebank. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 16–25, Barcelona, Spain (Online). Association for Computational Linguistics.

Randall K Barry. 1997. ALA-LC romanization tables-transliteration schemes for non-roman scripts. In *Library of Congress, 1997*.

Eckhard Bick and Tino Didriksen. 2015. CG-3 — beyond classical constraint grammar. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 31–39, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Jayeol Chun, Na-Rae Han, Jena D. Hwang, and Jinho D. Choi. 2018. Building Universal Dependency treebanks in Korean. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Clear Bible. 2022. MACULA Hebrew linguistic datasets.

Aslı Kuzgun, Oğuz Kerem Yıldız, Neslihan Cesur, Büşra Marşan, Arife Betül Yenice, Ezgi Sanıyar, Oguzhan Kuyrukçu, Bilge Nas Arıcan, and Olcay Taner Yıldız. 2021. From constituency to UD-style dependency: Building the first conversion tool of Turkish. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 761–769, Held Online. INCOMA Ltd.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

W.T. van Peursen, C. Sikkel, and D. Roorda. 2015. Hebrew text database ETCBC4b.

Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. 2017. Udapi: Universal API for Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 96–101, Gothenburg, Sweden. Association for Computational Linguistics.

Daniel Swanson, Tino Didriksen, and Francis M. Tyers. 2023. WITH context: Adding rule-grouping to VISL CG-3. In *Proceedings of the NoDaLiDa 2023 Workshop on Constraint Grammar - Methods, Tools and Applications*, pages 10–14, Tórshavn, Faroe Islands. Association of Computational Linguistics.

Daniel Swanson and Francis Tyers. 2022. A Universal Dependencies treebank of Ancient Hebrew. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2353–2361, Marseille, France. European Language Resources Association.

Graph Databases for Fast Queries in UD Treebanks

Niklas Deworetzki[1] **and Peter Ljunglöf**[1,2]
[1]Department of Computer Science and Engineering,
Chalmers University of Technology and University of Gothenburg
[2]Språkbanken Text, University of Gothenburg
nikdew@chalmers.se, peter.ljunglof@gu.se

Abstract

We investigate if labelled property graphs, and graph databases, can be a useful and efficient way of encoding UD treebanks, to facilitate searching for complex syntactic phenomena.

We give two alternative encodings of UD treebanks into the off-the-shelf graph database Neo4j, and show how to translate syntactic queries into the graph query language Cypher.

Our evaluation shows that graph databases can improve query times by several orders of magnitude, compared to existing approaches.

1 Motivation

Universal Dependencies (UD; de Marneffe et al., 2021) has celebrated 10 years of existence and has become a mature framework for text annotation. Currently there are almost 300 UD treebanks for almost 170 languages (Zeman et al, 2024).

One prominent use case of UD treebanks is to find examples of syntactic phenomena, within or across languages. E.g., Weissweiler et al. (2024) investigated if it is possible to identify grammatical constructions in different languages by searching for morphosyntactic patterns. Some of the queries they came up with were quite complex – they needed to cover all possible tree structures for a given construction, and at the same time rule out alternative interpretations.

There are several tools for searching in syntactic treebanks, such as ANNIS3 (Krause and Zeldes, 2014), AlpinoGraph (Kleiweg and van Noord, 2020), PML Tree Query (Štěpánek and Pajas, 2010), and Grew-match (Guillaume, 2021; Bonfante et al., 2018). They support complex queries and are usually efficient enough to be used on the existing UD treebanks. For example, Grew-match returns results within a few seconds, even when running a complex query on the largest UD treebank (Borges Völker et al., 2019).

If we are only interested in performing searches in manually annotated treebanks, the current tools are probably good enough. However, there are plenty of *automatically* annotated very large corpora.[1] If we want to perform searches for complex syntactic phenomena in such large treebanks (10–100 million tokens or more), the current query tools are not efficient enough. So there is a need for alternative approaches.

In this paper we investigate if existing off-the-shelf graph databases can be useful and efficient as a backend for complex searches in treebanks. We do this by giving two possible ways of encoding UD trees as *labelled property graphs*, which is the format underlying the Neo4j graph database (Francis et al., 2018). We also show how to translate Grew-match queries into the graph query language Cypher, and perform an extensive evaluation of the efficiency of both encodings, compared to each other, and to the Grew-match query system.

Our results show that existing off-the-shelf graph databases such as Neo4j can be very useful for performing large-scale complex syntactic searches in very large corpora.

1.1 Structure of the Paper

Section 2 gives an overview of UD treebanks, graph databases, and query languages. In section 3 we show two possible ways of encoding UD treebanks in a graph database, and in section 4 how to translate Grew-match queries into graph database queries. Section 5 consists of an evaluation of the two different treebank encodings, and in sections 6–7 we discuss the results and present some final conclusions.

[1] One such example is the research infrastructure Korp (Borin et al., 2012) from The Language Bank of Sweden, which contains more than 15 billion syntactically parsed and annotated tokens.

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 32–43
August 28-29, 2025 ©2025 Association for Computational Linguistics

2 Background

2.1 UD Treebanks as Graphs

Conceptually, a UD treebank consists of sentences where every sentence is a graph. The nodes in a graph are the words in the sentence, and the dependency relations are labelled directed edges between the word nodes. Words are annotated with different attributes, such as *lemma*, *part-of-speech*, and *morphological features*. Sentence graphs are required to form a tree with one single word being the **`root`**, according to the annotation guidelines for universal dependencies (Zeman et al., 2023).

In addition to the strict tree structure, it is possible to add *enhanced* dependency relations to a sentence, which might turn the sentence into a general graph. In this paper we will not assume that a treebank only consists of trees, so the encoding that we introduce in section 3 will work on more generic "graph banks" as well as on UD treebanks.

The basic tokenisation level in UD is *syntactic* words, and not phonological or orthographic words. Some languages contract words, such as English "isn't" (*is not*) and German "im" (*in dem*), and this can be encoded in a UD treebank using *multi-word tokens*. Conceptually, we can think of this as special kind of node, spanning multiple words.

2.2 Searching in UD Treebanks

A common system for graph-based searches in treebanks is Grew-match. Searching is done by sending a request containing multiple items, describing constraints on graphs. The main item is specified using the keyword **`pattern`** and contains a list of clauses, describing the nodes returned by a search. The **`with`** keyword is used to introduce clauses without adding additional nodes to the result returned for a request. The **`without`** keyword describes negative constraints – only graphs that do *not* match these are returned for a request.

A *clause* in Grew-match is either a node or edge declaration, or an additional constraint.

- A node declaration `X [attr="val"]` describes a node named `X` having a property `attr` with the value `val`. All nodes represent words and properties represent the feature structures on these words.
- An edge declaration `X -[rel]-> Y` connects two nodes named `X` and `Y`. This declaration requires that there is a dependency of type `rel` from the word `X` to the word `Y`.
- Additional constraints can express a certain word order. Writing `X < Y` requires that the word `X` *immediately* precedes the word `Y`. Writing `X << Y` requires that the word `X` occurs *somewhere* before the word `Y`.

Grew-match considers each sentence in a treebank as an individual graph and filters for those graphs matching the request.

2.3 Graph Databases

Neo4j is a general-purpose database system for graph-based data, similar to what a relational database is for tabular data. The system consists of a front-end in which you can formulate graph queries, a query-engine which plans and optimises the execution of queries, and a back-end storage system which handles persistence and data access. In contrast to Grew-match, Neo4j considers the whole database as a single graph.

The data model used by Neo4j is the *labelled property graph*, which represents data as a directed graph, where both nodes and edges may carry *labels* and attribute-value *properties*. This provides great flexibility and expressivity, as data can be represented in different nuanced ways.

In our encoding in section 3 we will use the labels mainly to specify the type of the node or label. Thus, when we write "a **`Word`** node", or "a **`SUCCESSOR`** edge", we actually mean "a node with the label **`Word`**", and "an edge with the label **`SUCCESSOR`**", respectively.

We adopt the convention that node labels are capitalised (e.g., **`Word`** and **`Sentence`**), but edge labels are uppercased (e.g., **`SUCCESSOR`** and **`DEPREL`**).

2.4 Cypher Query Language

The Cypher query language is used to query a Neo4j database (Francis et al., 2018). A query consists of multiple clauses, with the **`MATCH`**, **`WHERE`** and **`RETURN`** clauses being relevant for searching.

The **`MATCH`** clause introduces patterns to be matched in the graph. Writing `(n:Node {p:val})` in a **`MATCH`** clause describes a node named *n* labelled **`Node`** that has a property **`p`** with value `val`. An edge is written as `-[r:EDGE {p:val}]->` between two nodes. This example describes a directed edge labelled **`EDGE`** that has the property **`p`** with value `val` and is bound to the identifier *r*. In both cases, identifier, labels and properties are optional and may be omitted. The direction of an arrow describes the direction of a relationship.

```
ID   FORM    LEMMA   UPOS  XPOS  FEATS                                    HEAD  DEPREL
1    Surfen  Surfen  NOUN  NN    Gender=Neut|Number=Sing                    0   root
2-3  im      _       _     _     _                                          _   _
2    in      in      ADP   APPR  Case=Dat                                   4   case
3    dem     der     DET   ART   Case=Dat|Gender=Masc,Neut|Number=Sing      4   det
4    Garten  Garten  NOUN  NN    Gender=Masc|Number=Sing                    1   nmod
```

Figure 1: Example sentence (simplified) in CoNLL-U format from German-HDT (Borges Völker et al., 2019)

An edge suffixed with a plus character + indicates that two nodes are related via a sequence of edges matching the pattern (e.g. `-[:EDGE]->+` indicating a sequence of edges labelled **EDGE**).

The **WHERE** clause is used to additionally filter the matched subgraphs, using predicates that cannot be expressed by pattern matching. For example, properties of nodes and edges can be compared against each other or against regular expressions. In addition, there are keywords **EXISTS** and **NOT EXISTS** which have similar meaning as the **with** and **without** keywords in Grew-match.

Lastly, the **RETURN** clause is used to specify the result of a query. For every subgraph matched by a query, a record with all values specified in the **RETURN** clause will be returned.

While queries in Cypher are read from top to bottom with identifiers in later clauses being able to refer back to prior patterns, it is important to keep in mind that no order of execution is specified using queries. A database executing a query is free to reorder or simplify parts of the query in an attempt to optimize how it is executed, as long as the query result remains unchanged.

3 Encoding UD as Property Graphs

In order to store a UD-annotated treebank in a Neo4j database, it first has to be encoded as a labelled property graph. In this section we present an encoding scheme for dependencies, word annotations and structures, which we call the *property-based* encoding. Then we discuss an alternative encoding for annotations, which will be called the *node-based* encoding. Finally, we discuss how database constraints and indexes can be used to support the encoding.

3.1 Words and Dependencies

We encode each word as a node with label **Word**. A dependency between words is encoded as a **DEPREL** edge between the two corresponding **Word** nodes. The actual dependency relation is encoded as an edge property for the attribute **deprel**.

The root node in a sentence is encoded by labelling the corresponding **Word** node with the additional label **Root**.

3.2 Property-Based Encoding of Annotations

Figure 1 shows an example sentence in CoNLL-U format (Zeman et al., 2025).

The columns *ID*, *HEAD* and *DEPREL* (and *DEPS*, not shown in the example) are used to encode the (enhanced) dependency relation as **DEPREL** edges as discussed above, and therefore will not be encoded as node properties.

The columns *FORM*, *LEMMA*, *UPOS* and *XPOS* have a single value and will therefore each be encoded as single attributes to the **Word** node: **form**, **lemma**, **upos** and **xpos**, respectively. The column *FEATS* (and *MISC*, not shown in the example) contain attribute-value pairs. Each of these pairs will be encoded as an individual property.

As an example, wordline 3 in the example has a total of seven attributes to be encoded: *FORM=dem*, *LEMMA=der*, *UPOS=DET*, *XPOS=ART* are specified in separate columns, and we therefore straightforwardly encode them as properties on the corresponding **Word** node. The *FEATS* column specifies three morphological features: *Case=Dat*, *Gender=Masc,Neut* and *Number=Sing*, which are encoded as additional properties on the **Word** node.

3.3 Sentences

Sentences are annotated with metadata and span multiple tokens. To make this metadata accessible, we need a way to encode sentences and associate them with spanned tokens. We do this by introducing a **Sentence** node for every sentence and encoding its metadata as properties on that node.

To associate words with their **Sentence** node, we create a **DEPREL** edge with **deprel=root** towards the root node. Then all words will be connected to their **Sentence** node by following the **DEPREL** edges.

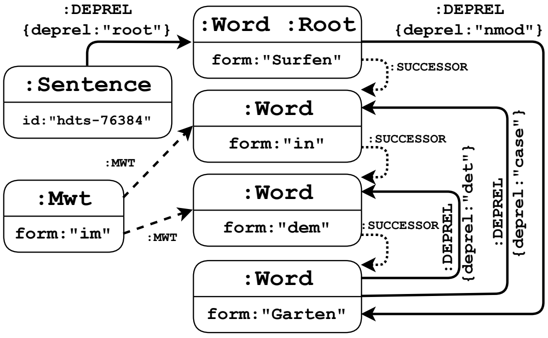


Figure 2: Example from Figure 1 encoded as a graph. Out of all properties, only **form** is shown.

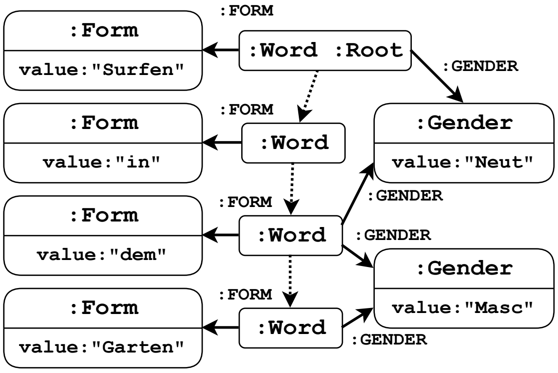


Figure 3: *Form* and *Gender* from Figure 1 encoded as a graph using the node-based encoding scheme.

To encode paragraphs or documents we follow the same strategy: create a **Paragraph** (or **Document**) node, and then create edges to its sentences (and edges from each document node to its paragraphs).

3.4 Linear Order

Words within a sentence are ordered, and a simple way to encode order in a graph database is via directed edges. Therefore, we introduce special edges to explicitly encode the word order within sentences.

For each word in a sentence we add a **SUCCESSOR** edge to its immediately succeeding word, except the final word which do not have a successor.

Figure 2 shows the example encoded as a labeled property graph, where **SUCCESSOR** edges are dotted. To reduce clutter only the **form** property is shown.

3.5 Multiword Tokens

The line "2–3" in Figure 1 is an example of a multiword token (MWT) – in German "im" is interpreted as a contraction of the syntactic words "in dem".

We encode multiword tokens in a similar way to sentences. For each multiword token, we add a new **Mwt** node, and **MWT** edges (dashed in Figure 2) from the node to each spanned word.

3.6 Alternative, Node-Based Encoding

In section 3.2 we showed how to encode attribute values as properties directly on the **Word** nodes.

An alternative strategy is to create a new node for each value of an attribute for a word, and add an edge from the **Word** node to the attribute node. For example, the attribute *Gender=Masc* would be represented as a **Gender** node, annotated with the property **value=Masc**. Note that we only create one such **Gender** node with the value **Masc**, and it will be shared by all **Word** nodes.

There are multiple possible advantages of this encoding strategy. As following edges between nodes is a fast operation when querying a graph database, this encoding scheme could lead to better query performance. Further, by reusing nodes we aim to deduplicate data. Sections 5.1 and 5.4 show the impact on encoding size and query speed.

Further, wordline 3 in Figure 1 has a property *Gender=Masc,Neut*, meaning that the determiner can act as either masculine or neuter. This can be encoded as two **Gender** edges from the word, to the **Masc** and **Neut** nodes respectively. This can simplify some queries quite a lot, but we have not looked into this because it is not needed for our translation from the Grew-match query language.

Figure 3 shows the node-based encoding of the example in Figure 1. To reduce clutter only the **Form** and **Gender** nodes are shown.

The node- and the property-based encodings are freely interchangeable. The encoding of dependency relations, sentences and MWTs remains unchanged when choosing between the node-based and the property-based encoding strategy. It is even possible to mix both strategies, encoding only some attributes as nodes and others as properties.

3.7 Constraints and Indexes

Adding constraints and indexes helps the query planner to improve query performance. Uniqueness constraints in Neo4j ensure that a combination of label and property value appears only once in the database. Similarly, indexes can be used to quickly find nodes or edges with a combination of label and property value.

We add a uniqueness constraint and index for every attribute encoded using the node-based strategy. This should improve performance for queries on the node-based encoding. Additionally, we add another index on **DEPREL** edges and their **deprel** property, which enables fast lookup of nodes connected by an edge in the index.

4 Querying in Encoded Corpora

To demonstrate the capabilities of the Cypher query language, we will now describe a straightforward algorithmic approach to translate Grew-match queries into Cypher queries.

For every word matched in a Grew-match **pattern**, add a **MATCH** clause for a **Word** node with the same identifier. Further, add a single **RETURN** clause at the end of the query and add all introduced identifiers to that clause. While it is not necessary to match all words first, doing so simplifies the translation, as words can be referred to by their identifier afterwards. Then, translate each of the clauses in a Grew-match request as follows:

- An edge clause specifying a dependency relation `X -[aux]-> Y`, is translated into a **MATCH** clause specifying the same edge `(X)-[:DEPREL {deprel:"aux"}]->(Y)`.
- Clauses specifying immediate precedence (written `X < Y`), are translated into a **MATCH** clause with an edge `(X)-[:SUCCESSOR]->(Y)`. General precedence between nodes (written `X≪Y`) is translated similarly, allowing the two nodes to be related via a sequence of edges `(X)-[:SUCCESSOR]->+(Y)`.
- **with** clauses are translated as if they were part of a **pattern**, adding an initial **MATCH** clause for all occurring words and translating each clause. Identifiers for these words must not be included in the **RETURN** clause and possibly require renaming if they occur in multiple **with** patterns.
- **without** clauses are translated into a **NOT EXISTS** expression as part of the **WHERE** clause containing a translation of the individual Grew-match clauses.

Clauses in Grew-match accessing annotations have to be translated differently depending on the encoding scheme. We will consider how this is done on the example `X [upos="NOUN"]`.

- If encoded as properties, a **MATCH** clause is added specifying identifier and the requested properties: `(X {upos:"NOUN"})`.
- If encoded as nodes, a **MATCH** clause is added for an edge between the word node and the node representing the requested value: `(X)-[:UPOS]->(:Upos {value:"NOUN"})`. If multiple attribute-value pairs are specified, a **MATCH** clause is added for each of them.

Values of properties in Grew-match can also be specified in terms of a regular expression or a disjunction of values. In these cases a direct translation into a **MATCH** clause is not possible and we use the **WHERE** clause to represent these constraints. For the node-based encoding of annotations, the corresponding nodes have to be fetched via a **MATCH** clause without specifying their value. A Grew-match clause like `X [lemma="der"|"die"]` therefore turns into `WHERE X.lemma IN ["der","die"]`, under the property-based annotation scheme. For the node-based annotation scheme it is instead translated into the following two clauses: `MATCH (X)-[:LEMMA]->(xlemma:Lemma)`, and `WHERE xlemma.value IN ["der","die"]`.

There are some special cases to consider when translating queries. A query for the root node of a sentence can be translated into a **MATCH** clause with the **Root** label. Further, queries in Grew-match consider each sentence as an individual graph. Consequently, all words in a Grew-match request are implicitly constrained to the same sentence. This restriction has to be translated as well, by adding a **MATCH** clause relating otherwise unrelated words to the same dependency tree: `(X)-[:DEPREL]-+(Y)`.

5 Evaluation

This section presents the performance of Neo4j as a corpus system for UD-annotated treebanks in different scenarios to evaluate, (a) whether Neo4j is a viable system for treebanks and (b) how the presented encoding schemes perform. We mainly consider two perspectives here: The perspective of an administrator encoding treebanks and provisioning the storage space for the database. And the perspective of users wanting to search treebanks with quick query response times. Our evaluations cover encoding time, required disk space for encoded treebanks and query execution time.

We developed a tool to automatically encode and import UD-annotated treebanks into a Neo4j database using the presented encoding schemes. The tool and its source code is freely accessible online.[2] Our tool accepts CoNLL-U files as its

[2] Source code, executables and experiment data is available at https://github.com/Niklas-Deworetzki/neo4j-ud-importer

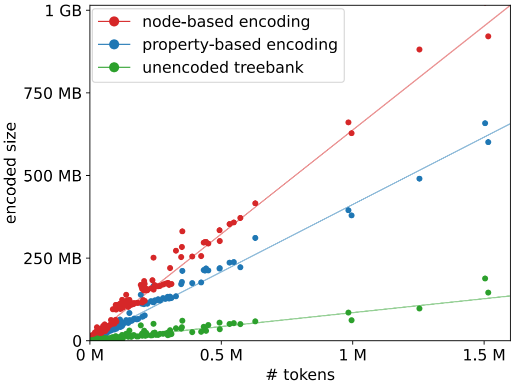


Figure 4: Disk size of treebanks.

input, encodes the treebank described by these files as a graph and stores this graph in a Neo4j database.

Our measurements were obtained using Neo4j version 5.26.0 and Grew-match version 1.16.1, both running in Docker on the same hardware.

5.1 Encoding Corpora

We automatically encoded all treebanks in UD release 2.15 (Zeman et al, 2024) to measure time and disk space requirements. Figure 4 shows the required disk size of treebanks in bytes in relation to treebank size in tokens.[3] The disk size of an encoded treebank is calculated as the sum of the size of all files in the `database` directory in Neo4j. A clear linear relationship between disk size and treebank size can be seen. The encoding time also has a strong linear relationship with treebank size – it takes around 1 minute for the property-based encoding to encode a 1-million token treebank, and 2–3 minutes for the node-based encoding. The property-based encoding requires approximately 6 times as much disk space as the CoNLL-U files, while the node-based encoding requires approximately 10 times as much disk space.

5.2 Benchmarking Setup

To measure the performance of Neo4j as a query system for UD-annotated treebanks, we translated and ran a set of queries from Weissweiler et al. (2024). They present rules for the Grew-match graph rewriting framework to automatically annotate constructions in UD treebanks. We selected queries from these rules for four different constructions present in ten different languages (namely interrogatives, existentials, conditionals and NPN – a repeated noun with an adposition in between).

Important for our selection is that the chosen requests cover a variety of languages, cover many aspects of the Grew-match query language, and are relevant for linguistic research. Details of these queries are not important to our evaluation, but are explained further in Appendix C. We used the procedure from section 4 to translate the four chosen patterns for each of the different languages into equivalent Cypher queries for both encoding variants, resulting in a total of 80 translated queries.

To execute queries and measure their execution time, we grouped queries for the same language and encoding scheme together. The four queries were executed in sequence, and the sequence was repeated multiple times. The goal of this is to increase cache pressure, so that it is not possible for a query system to simply “remember” the results for one particular query. We then started up a server with one encoded corpus, executed the sequence of queries for that corpus 10 times to warm up caches, and then took measurements by repeating queries in sequence 100 times. For each query, we recorded the median of all 100 collected execution times. Queries for Neo4j were sent to the database server for execution, while queries for Grew-match were executed by accessing its command line interface.

We selected the biggest available corpus for each of the ten languages to run our measurements on. The complete list of languages and used corpora can be found in Appendix A.

5.3 Comparing Neo4j and Grew-match

Neo4j using the property-based encoding requires on average 1% of query execution time compared to Grew-match. More precisely, Grew-match requires between 30 (for the NPN query on the Hindi treebank) and 600 (for conditionals in Portugese and existentials in Spanish) times as much execution time. There is, however, one exception: for the interrogatives in Hindi, Neo4j was actually slower, requiring 10% more execution time.

On all measured systems, the execution time is roughly proportional to the size of the queried treebank. Per million tokens of treebank size, Grew-match requires on average 28 seconds of query time, while Neo4j requires 0.28 seconds (for the property-based encoding) and 0.31 seconds (for the node-based encoding). A table listing all execution times is shown in Appendix B.

[3] The figure omits the Hamburg Dependency Treebank, which with 3.4 million tokens lies far outside the shown range and approximately 5% below the trend line.

Language	Cond.	Exist.	Interrog.	NPN
Chinese	0.24	0.72	0.25	1.60
Coptic	0.67	0.85	0.66	1.94
English	0.55	0.52	0.70	2.64
French	0.59	1.05	**5.45**	1.24
German	0.64	0.58	0.08	1.33
Hebrew	0.62	0.78	**5.99**	0.98
Hindi	0.27	0.39	1.07	**6.76**
Portuguese	0.88	0.57	0.21	1.57
Spanish	0.51	0.74	0.16	1.47
Swedish	0.67	0.50	0.74	1.52
Average	0.53	0.64	0.35	1.53

Table 1: Execution time of the node-based encoding, relative to the property-based encoding, per query type. A value of 0.25 means that the node-based encoding is 4 times faster. Outliers are **bold-faced**, and they are *not* included in the calculation of the average speed-up. Because the values are factors, the average is calculated as the geometric mean.

5.4 Comparing Encoding Strategies

A comparison of the query execution times for the property-based and node-based encoding is shown in Table 1. The results in that table show groups of similar relative execution times: In general, the node-based encoding is faster, requiring 40% to 80% of execution time for most queries. For interrogatives in Hindi and NPN's in Hebrew, there is no difference between both encoding schemes. For all of the NPN queries, the node-based encoding is actually slower, requiring 1.5 times longer execution time on average. For interrogatives in French and Hebrew, as well as NPN's in Hindi, the node-based encoding is 5–7 times slower. On the other hand, it is 4–10 times faster for conditionals in Hindi and Chinese, and for interrogatives in Chinese, German, Portuguese and Spanish.

5.5 Execution Time and Corpus Size

To better understand how execution times scale with respect to corpus size, we ran the same set of queries on differently-sized subsets of the Hamburg Dependency Treebank. These sub-corpora were obtained by randomly sampling $10\%, 20\%, \ldots, 90\%$ of sentences from the original corpus.

We used the same setup as presented in Section 5.2 to execute all four queries for the German language on these corpora. The measured execution times are shown in Figure 5 and show a clear linear relationship between execution time and corpus size for each of the executed queries.

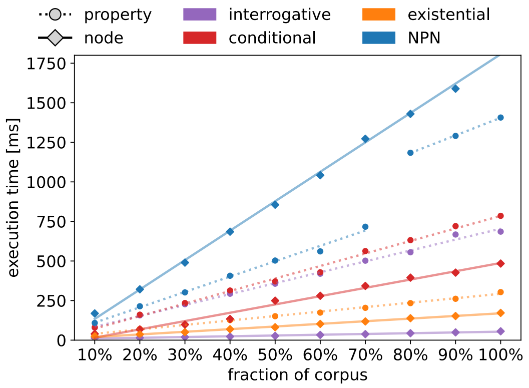


Figure 5: Execution time for differently sized sub-corpora of HDT. Note the discontinuity for NPN.

There is one exception to this linear dependency, which is seen in the diagram: there is an unexpected jump for NPN queries in the property-based encoding after 70% of the corpus size.

6 Discussion

Our experiments clearly show that Neo4j is a viable corpus search system for UD treebanks.

6.1 Comparing Neo4j and Grew-match

On average, Neo4j outperforms Grew-match in almost all cases by orders of magnitude. Most queries run in fractions of a second where Grew-match needs several seconds for the same query.

We believe that one reason for this improvement is that Neo4j considers the whole treebank as one single graph, and therefore it can make use of search indexes (as discussed in section 3.7). Grew-match on the other hand considers every sentence to be a separate graph, which makes it much harder to do global optimisations, and therefore it has to test each sentence against the query iteratively.

The main trade-off with using Neo4j instead of Grew-match is disk usage. Figure 4 shows that encoding the treebank in a graph database uses 6 – 10 times more space than the original CoNLL-U text files, depending on the encoding.

For example, all UD treebanks combined consist of roughly 32M tokens, or 2.9 GB of CoNLL-U text files. In comparison, the Neo4j database files require 14 or 22 GB (depending on the encoding), which is a considerable overhead, but still manageable.

6.2 Hindi Interrogatives

There is one notable exception, the interrogative query in Hindi shows no improvement at all compared to Grew-match. This is the case both for the node-based and the property-based encoding. The query itself consists of a disjunction of several lemmas, followed by a filter that rules out sentences containing some subtrees that are unrelated to the actual lemma. Because of this, Neo4j has to add additional constraints to make sure that the subtrees are in the same sentence as the lemma it is searching for, so it has to do extra work and cannot make use of the global indexes. See Appendix C for the query and its translation.

We did try a simple optimisation in our encodings, where we created direct edges from word nodes to their sentence nodes. This improved the execution time for Hindi interrogatives (and some other queries) by up to 100 times. We did not perform any in-depth evaluation of this and other possible optimisations, but it suggests that it is possible to improve the corpus encoding substantially if we know what kind of queries we will perform.

6.3 Comparing Encodings

The size of the encoded corpora grow linearly in the size of the treebanks, and the property-based encoding requires only around 60% as much storage as the node-base encoding. Extracting different values into separate, shared nodes provides no benefit in terms of storage size. The reason for this is that Neo4j stores string values not as part of nodes, but in a separate unit (Rocha, 2020). Therefore, strings occurring multiple times in the dataset will result in only one copy stored in the database with multiple references to that one copy.

In terms of execution time, the node-based encoding is usually faster than the property-based, by a factor of 1.5–3. But this is not always the case: for the NPN queries it is the property-based encoding that is faster by a factor of 1.5–2.5. And there are some few extreme outliers, where the property-based encoding is actually 6–7 times faster.

Looking into the execution plans and profiling information for these queries suggests that having each attribute as a separate node in the graph is the reason for both of these behaviors. In situations where the node-based encoding outperforms the property-based one, it does so by making use of uniqueness constraints and indexes. For example, one lookup in the `POS`-index will yield the node for a certain part-of-speech, which has a reference to all matching words. The property-based representation on the other hand, linearly scans through words to find nodes for which relations can be resolved and further constraints checked. When it comes to the NPN construction queries, this linear scan is advantageous. The query asks for three subsequent tokens, and the database has the successors readily available when we use the property-based encoding. For the node-based encoding, the database opts to find all words related to the single `Noun` node, followed by subsequently finding their successors and their part-of-speech nodes. This results in many accesses to the underlying storage at many different positions, resulting in slow execution times.

The conclusion from this is, that the node-based encoding can make use of available indices and uniqueness constraints efficiently, outperforming the property-based encoding for most queries. However, this is not true in all cases, and the relative simplicity of the property-based encoding sometimes results in lower execution times, as shown by the NPN queries.

6.4 Scalability

Comparing the same query on differently sized sub-corpora we see that the execution time grows linearly in the size of the corpus size for all queries and encoding strategies. We do not know why the property-based encoding experiences a bigger-than-expected jump in execution time for NPN queries when going from 70% to 80% of the original corpus size. Maybe it has to do with caching of intermediate results and that the system runs out of internal memory, but that is just a guess.

Encoding the corpus in Neo4j seems to improve the search speed by around 100 times on average, compared to Grew-match. Therefore we draw the conclusion that it should be feasible to use any of our encodings on treebanks with 100 million tokens or more. Such a treebank would require about 100 GB of storage space, which is feasible on modern computers.

Note that we got these improvements despite using a very simplistic encoding of the treebanks into a graph database. As suggested by our optimisation in section 6.2 there is probably a lot of opportunities for further improvement.

6.5 Use as a Corpus System

One thing we have not looked into in this study is how to incorporate Cypher and Neo4j in a full-fledged corpus system, such as Grew. Grew-match is just one part of the Grew system, which is a general graph-rewriting framework with which one can create, annotate, and update treebanks. In addition to searching within a treebank, nodes and edges can be created or deleted, nodes can be re-ordered and annotations can be changed. Cypher supports similar functionality via commands such as **CREATE**, **DELETE**, and **SET**, for modifying the database in different ways. More work would be required to map different Grew commands to these Cypher clauses.

7 Conclusion

Our main conclusion from this evaluation is that graph databases are viable as backend storage for treebanks. The study is only done on UD treebanks, but there is nothing very UD-specific in our encodings or the graph databases, so we believe that this would be useful for all kinds of treebanks.

Using graph databases it will be possible to search for complex syntactic phenomena in very large treebanks with 100 million tokens and more.

Since we translate the treebanks to a general graph, it should definitely be possible to include more kinds of relations, such as anaphoric references, semantic databases, and morphological segmentation. Including all kinds of relations in one single graph database opens up for doing large-scale searching for complex queries on several linguistic levels at once.

Acknowledgments

We thank Nicholas Smallbone and three anonymous reviewers for valuable discussions, comments and insights.

References

Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, et al. 2017. The Hindi/Urdu treebank project. In *Handbook of Linguistic Annotation*. Springer Press.

Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*. Wiley Online Library.

Emanuel Borges Völker, Maximilian Wendt, Felix Hennig, and Arne Köhn. 2019. HDT-UD: A very large Universal Dependencies treebank for German. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 46–57, Paris, France. Association for Computational Linguistics.

Lars Borin, Markus Forsberg, and Johan Roxendal. 2012. Korp – the corpus infrastructure of Språkbanken. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, pages 474–478, Istanbul, Turkey. European Language Resources Association (ELRA).

António Branco, João Ricardo Silva, Luís Gomes, and João António Rodrigues. 2022. Universal grammatical dependencies for Portuguese with CINTIL data, LX processing and CLARIN support. In *Proceedings of the 13th International Conference on Language Resources and Evaluation (LREC'22)*, pages 5617–5626, Marseille, France. European Language Resources Association (ELRA).

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

Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. 2018. Cypher: An evolving query language for property graphs. In *Proceedings of the 44th International Conference on Management of Data (SIGMOD'18)*, page 1433–1445, Houston, USA. Association for Computing Machinery (ACM).

GSDSimp. 2023. Simplified chinese universal dependencies. Accessed 2025-04-14, https://github.com/UniversalDependencies/UD_Chinese-GSDSimp.

Bruno Guillaume. 2021. Graph matching and graph rewriting: GREW tools for corpus exploration, maintenance and conversion. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL): System Demonstrations*, pages 168–175, Online. Association for Computational Linguistics (ACL).

Bruno Guillaume, Marie-Catherine de Marneffe, and Guy Perrier. 2019. Conversion et améliorations de corpus du français annotés en Universal Dependencies [conversion and improvement of Universal Dependencies French corpora]. *Traitement Automatique des Langues*, 60(2):71–95.

Peter Kleiweg and Gertjan van Noord. 2020. AlpinoGraph: A graph-based search engine for flexible and efficient treebank search. In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, pages 151–161, Düsseldorf, Germany. Association for Computational Linguistics.

Thomas Krause and Amir Zeldes. 2014. ANNIS3: A new architecture for generic corpus query and visualization. *Digital Scholarship in the Humanities*, 31(1):118–139.

Taulé Mariona, M. Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC'08)*, pages 96–101, Marrakech, Morocco. European Language Resources Association (ELRA).

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Martha Palmer, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17.

José Rocha. 2020. Understanding Neo4j's data on disk. Accessed 2025-04-14, https://neo4j.com/developer/kb/understanding-data-on-disk/.

Shoval Sade, Amit Seker, and Reut Tsarfaty. 2018. The Hebrew Universal Dependency treebank: Past, present and future. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jan Štěpánek and Petr Pajas. 2010. Querying diverse treebanks in a uniform way. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Leonie Weissweiler, Nina Böbel, Kirian Guiller, Santiago Herrera, Wesley Scivetti, Arthur Lorenzi, Nurit Melnik, Archna Bhatia, Hinrich Schütze, Lori Levin, Amir Zeldes, Joakim Nivre, William Croft, and Nathan Schneider. 2024. UCxn: Typologically informed annotation of constructions atop Universal Dependencies. In *Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING'24)*, pages 16919–16932, Torino, Italia. European Language Resources Association (ELRA).

Amir Zeldes and Mitchell Abrams. 2018. The Coptic Universal Dependency treebank. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 192–201, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, Nathan Schneider, Filip Ginter, and Christopher Manning. 2023. Universal dependencies guidelines. Accessed 2025-04-14, https://universaldependencies.org/guidelines.html.

Daniel Zeman, Joakim Nivre, Nathan Schneider, Filip Ginter, and Sampo Pyysalo. 2025. CoNLL-U format. Accessed 2025-04-14, https://universaldependencies.org/format.html.

Daniel Zeman et al. 2024. Universal Dependencies 2.15. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

A List of Treebanks used for Execution Time Measurements

The complete list of all treebanks used for our execution time measurements is shown in Table 2. The collection of languages is determined by the languages for which Weissweiler et al. (2024) provide automated annotation rules. We chose the largest available treebank for each of these languages.

B Table of Execution Time Measurements

Table 3 contains the results of our benchmark, showing all 120 data points obtained from running queries for 4 different constructions in 10 different languages on 3 query systems. The order of languages presented in this table follows Table 2.

C List of Grew-match and Neo4j Queries

A selection of queries used for our execution time measurements is shown in Table 4. The complete list of queries is available online: https://github.com/Niklas-Deworetzki/neo4j-ud-importer/tree/main/experiments/queries

Lang.	**Treebank**	**# Tokens**
German	HDT (Borges Völker et al., 2019)	3,399,390
Spanish	AnCora (Mariona et al., 2008)	547,558
Portuguese	CINTIL (Branco et al., 2022)	441,991
French	GSD (Guillaume et al., 2019)	389,367
Hindi	HDTB (Bhat et al., 2017; Palmer et al., 2009)	351,704
English	EWT (Silveira et al., 2014)	251,493
Chinese	GSDSimp (GSDSimp, 2023)	123,291
Hebrew	HTB (Sade et al., 2018)	114,648
Swedish	Talbanken (Nivre et al., 2006)	96,820
Coptic	Scriptorium (Zeldes and Abrams, 2018)	26,837

Table 2: UD treebanks used for our benchmark ordered by size.

Lang.	**Query**	**Grew**	**Prop.**	**Node**	**Lang.**	**Query**	**Grew**	**Prop.**	**Node**
German	cond.	70.84	0.714	0.458	**Spanish**	cond.	15.05	0.059	0.030
	exist.	67.49	0.283	0.165		exist.	14.95	0.025	0.019
	interrog.	67.43	0.643	0.050		interrog.	15.33	0.126	0.020
	NPN	68.02	1.282	1.699		NPN	15.05	0.179	0.262
Portugese	cond.	8.52	0.014	0.012	**French**	cond.	7.18	0.035	0.021
	exist.	8.50	0.103	0.059		exist.	7.16	0.019	0.020
	interrog.	8.48	0.089	0.019		interrog.	7.26	0.081	0.441
	NPN	8.57	0.135	0.212		NPN	7.21	0.151	0.188
Hindi	cond.	16.16	0.109	0.029	**English**	cond.	5.79	0.049	0.027
	exist.	16.24	0.224	0.086		exist.	5.77	0.064	0.033
	interrog.	16.22	17.633	18.906		interrog.	5.84	0.028	0.020
	NPN	16.19	0.028	0.192		NPN	6.00	0.043	0.112
Chinese	cond.	3.54	0.033	0.008	**Hebrew**	cond.	3.11	0.027	0.017
	exist.	3.53	0.020	0.014		exist.	3.09	0.017	0.013
	interrog.	3.52	0.029	0.007		interrog.	3.11	0.020	0.119
	NPN	3.54	0.041	0.065		NPN	3.12	0.094	0.092
Swedish	cond.	2.43	0.031	0.021	**Coptic**	cond.	1.28	0.021	0.014
	exist.	2.42	0.031	0.015		exist.	1.27	0.011	0.009
	interrog.	2.45	0.019	0.014		interrog.	1.29	0.021	0.014
	NPN	2.43	0.042	0.063		NPN	1.32	0.017	0.033

Table 3: Execution times measured in seconds for the three query systems (**Grew**-match, Neo4j with **prop**erty-based encoding and Neo4j with **node**-based encoding) on equivalent queries ordered by language and construction.

Query	Grew-match	Property-based	Node-based

German exist.	`pattern {`
	`  E[lemma="es"];`
	`  G[lemma="geben"];`
	`  G-[nsubj]->E;`
	`}`
		`MATCH (E:Word)`
		`MATCH (G:Word)`
		`MATCH (E {LEMMA:’es’})`
		`MATCH (G {LEMMA:’geben’})`
		`MATCH (G)-[:DEPREL`
		`  {deprel:’nsubj’}]->(E)`
		`RETURN E, G`
			`MATCH (E:Word)`
			`MATCH (G:Word)`
			`MATCH (E)-[:LEMMA]->`
			`  (:Lemma {value:’es’})`
			`MATCH (G)-[:LEMMA]->`
			`  (:Lemma {value:’geben’})`
			`MATCH (G)-[:DEPREL`
			`  {deprel:’nsubj’}]->(E)`
			`RETURN E, G`

Hindi interrog.	`pattern {`
	`  W [lemma="क्या"`
	`   |"कौन"`
	`   |"कहां"`
	`   |"कब"`
	`   |"कैसे"`
	`   |"कितना"`
	`   |"किस"];`
	`} without {`
	`  SC [form="कि"];`
	`  V -[mark]-> SC`
	`} without {`
	`  V1 [upos=VERB];`
	`  V1 -[advcl]-> V;`
	`}`
		`MATCH (W:Word)`
		`WHERE W.LEMMA in [...]`
		`AND NOT EXISTS {`
		`  MATCH (SC:Word)`
		`  MATCH (V:Word)`
		`  MATCH (V)-[:DEPREL]-+(W)`
		`  MATCH (SC FORM:’कि’)`
		`  MATCH (V)-[:DEPREL`
		`   {deprel:’mark’}]->(SC)`
		`} AND NOT EXISTS {`
		`  MATCH (V1:Word)`
		`  MATCH (V:Word)`
		`  MATCH (V1)-[:DEPREL]-+(W)`
		`  MATCH (V1 UPOS:’VERB’)`
		`  MATCH (V1)-[:DEPREL`
		`   {deprel:’advcl’}]->(V)`
		`} RETURN W`
			`MATCH (W:Word)`
			`MATCH (W)-[:LEMMA]->`
			`  (wlemma:Lemma)`
			`WHERE wlemma.value in [...]`
			`AND NOT EXISTS {`
			`  MATCH (SC:Word)`
			`  MATCH (V:Word)`
			`  MATCH (V)-[:DEPREL]-+(W)`
			`  MATCH (SC)-[:FORM]->`
			`   (:Form {value:’कि’})`
			`  MATCH (V)-[:DEPREL`
			`   {deprel:’mark’}]->(SC)`
			`} AND NOT EXISTS {`
			`  MATCH (V1:Word)`
			`  MATCH (V:Word)`
			`  MATCH (V1)-[:DEPREL]-+(W)`
			`  MATCH (V1)-[:UPOS]->`
			`   (:Upos {value:’VERB’})`
			`  MATCH (V1)-[:DEPREL`
			`   {deprel:’advcl’}]->(V)`
			`} RETURN W`

Chinese NPN	`pattern {`
	`  N1 [upos=NOUN];`
	`  P [upos=ADP];`
	`  N2 [upos=NOUN];`
	`  N1 < P;`
	`  P < N2;`
	`  N1.form=N2.form;`
	`}`
		`MATCH (N1:Word)`
		`MATCH (P:Word)`
		`MATCH (N2:Word)`
		`MATCH (N1 {UPOS:’NOUN’})`
		`MATCH (N2 {UPOS:’NOUN’})`
		`MATCH (P {UPOS:’ADP’})`
		`MATCH (N1)-[:SUCCESSOR]->(P)`
		`MATCH (N2)<-[:SUCCESSOR]-(P)`
		`WHERE N1.FORM = N2.FORM`
		`RETURN N1, N2, P`
			`MATCH (N1:Word)`
			`MATCH (P:Word)`
			`MATCH (N2:Word)`
			`MATCH (N1)-[:UPOS]->`
			`  (:Upos {value: ’NOUN’})`
			`MATCH (N2)-[:UPOS]->`
			`  (:Upos {value: ’NOUN’})`
			`MATCH (P)-[:UPOS]->`
			`  (:Upos {value: ’ADP’})`
			`MATCH (N1)-[:SUCCESSOR]->(P)`
			`MATCH (N2)<-[:SUCCESSOR]-(P)`
			`WHERE (N1)-[:FORM]->(:Form)`
			`  <-[:FORM]-(N2)`
			`RETURN N1, N2, P`

Table 4: A sample of queries used for execution time measurements. The table shows a Grew-match pattern and the translated Cypher queries, following the translation scheme provided in Section 4. Note that the full list of lemmas for the Hindi interrogative query is only shown for Grew-match and is abbreviated for the other two columns.

STARK:
A Toolkit for Dependency (Sub)Tree Extraction and Analysis

Luka Krsnik
Centre for Language Resources and Technologies
University of Ljubljana
Ljubljana, Slovenia
krsnik.luka92@gmail.com

Kaja Dobrovoljc
University of Ljubljana
Jozef Stefan Institute
Ljubljana, Slovenia
kaja.dobrovoljc@ff.uni-lj.si

Abstract

We present STARK, a lightweight and flexible Python toolkit for extracting and analyzing syntactic (sub)trees from dependency-parsed corpora. By systematically slicing each sentence into interpretable syntactic units based on configurable parameters, STARK enables bottom-up, data-driven exploration of syntactic patterns at multiple levels of abstraction—from fully lexicalized constructions to general structural templates. It supports any CoNLL-U-formatted corpus and is available as a command-line tool, Python library, and interactive online demo, ensuring seamless integration into both exploratory and large-scale corpus workflows. We illustrate its functionality through case studies in noun phrase analysis, multiword expression identification, and syntactic variation across corpora, demonstrating its utility for a wide range of corpus-driven syntactic investigations.

1 Introduction

Syntactically annotated corpora, or treebanks, have become indispensable in linguistic research, supporting work on grammar description (Ferrer-i Cancho et al., 2022), typological comparison (Levshina, 2022), genre analysis (Wang and Liu, 2017), as well as language technology development and understanding (Zeman et al., 2018; Lin et al., 2012; Hewitt and Manning, 2019). As their availability grows, so too does the ecosystem of tools designed to facilitate their exploration, most notably through treebank browsing services such as Grew-match (Guillaume, 2021), PML Tree Query (Štepánek and Pajas, 2010), and INESS (Rosén et al., 2012).

Despite this growing infrastructure, most existing tools are inherently *query-based*. They require the user to formulate a specific hypothesis or structural pattern of interest—typically by specifying the number of words involved, their morphological properties, and their syntactic relationships. Such top-down approaches are well-suited for targeted investigation, but they offer limited support for inductive, bottom-up discovery of patterns—particularly in cases where no prior expectations about syntactic configurations are available or desirable.

In practical terms, if a researcher is interested in noun phrase structures, most existing tools will allow them to search for examples of a specific pattern — for example, a noun preceded by an adjectival modifier. However, such tools do not typically support asking what kinds of noun phrase structure patterns actually occur in the corpus, how frequent they are, or whether any rare or unexpected patterns emerge — particularly in contrast to another dataset.

To address this gap, we present STARK (Subtree Analysis and Retrieval Kit), a toolkit for bottom-up, treebank-driven syntactic analysis. Rather than relying on predefined queries, STARK automatically extracts all trees and subtrees that meet general structural criteria specified by the user—effectively slicing a parsed corpus into interpretable syntactic patterns, which can then be counted and compared within or across corpora.

The remainder of the paper introduces STARK's core functionality and configurable parameters (§2), illustrates its analytic capabilities through frequency, association, and comparison outputs (§3), and then details its features for example retrieval and visualization (§4), scalability and performance optimization (§5), and accessibility via an interactive online demo and open-source release (§6).

2 Core Functionality

STARK is an open-source Python toolkit for extracting and analyzing syntactic (sub)trees from dependency-parsed corpora. It operates by systematically slicing each sentence into smaller, interpretable syntactic units based on configurable structural parameters, described below.

2.1 Basic Design

STARK operates on input files in CoNLL-U format, the standard tab-separated format for representing word-level syntactic and morphological annotations in dependency-parsed corpora. Although the tool was developed with the Universal Dependencies (UD) annotation scheme (de Marneffe et al., 2021) in mind, it is not limited to UD-compliant data: it accepts any corpus in CoNLL-U format, regardless of scheme-specific tagsets or label inventories, and handles multi-root sentence structures and other non-canonical phenomena.

To illustrate STARK's core functionality, consider the sentence in Figure 1:

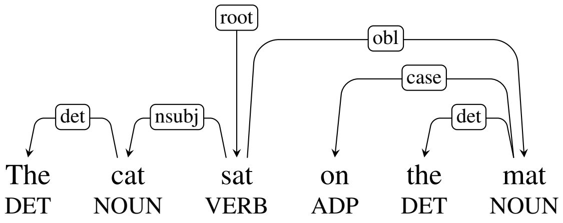


Figure 1: Dependency tree for the sentence *The cat sat on the mat* using the UD annotation scheme.

STARK treats every word in a sentence as a potential syntactic head and extracts the subtree rooted at that word—that is, the head and all its dependents. This yields a collection of overlapping (sub)trees,[1] each capturing a local syntactic configuration. Table 1 shows the resulting structures extracted using this procedure, if we were to extract unlabeled trees with surface word forms only (but see Section 2.2 for more options).

Head	**Subtree**
`The`	`The`
`cat`	`The < cat`
`sat`	`(The < cat) < sat > (on < the < mat)`
`on`	`on`
`the`	`the`
`mat`	`on < the < mat`

Table 1: (Sub)trees extracted from the parsed sentence in Figure 1.

Each unique tree is then counted and written to a tab-separated output file, one per row. Trees are represented using a simplified query-like syntax inspired by `dep_search` tool (Luotolahti et al., 2017),[2] with additional columns optionally showing frequency, statistical scores, or illustrative examples.

We now turn to the main parameters that control how trees are represented, filtered, and extracted.

2.2 Tree Representation

STARK offers several options for determining how trees are constructed and represented in the output.

- **Node type** (`--node_type`) determines what information is used to represent each token. Users can choose any CoNLL-U field, such as surface form, lemma, or UPOS tag, or omit node content entirely. For instance, the tree `the < mat` could appear as `DET < NOUN` (UPOS), or `the < mat` (lemma), depending on the setting.

- **Dependency labels** (`--labeled`) are typically included (e.g., `the <det mat`), but can also be omitted to support more abstract structure. Subtype retention is optional (`--label_subtypes`), allowing users to distinguish between coarse and fine-grained labels (e.g., `nmod:poss` vs. `nmod`).

- **Node order** (`--fixed`) determines whether linear word order is taken into account. When enabled, token order contributes to the identity of the tree; otherwise, trees are treated as equivalent regardless of surface order, which is particularly useful for analyzing languages with flexible constituent structure.

These settings allow users to extract trees at different levels of specificity, from fully lexicalized constructions (e.g., Table 3) to more abstract syntactic patterns (e.g., Table 2 and 4).

2.3 Tree Filtering

Users can further restrict extracted trees using a range of filters:

- **Tree size** (`--size`) specifies the number of nodes in each tree, either as a single value (e.g., 3) or a range (e.g., 2-5). This setting is optional—using a broad range like 1-1000 effectively extracts all trees, regardless of size.

[1] In what follows, we use the term *tree* to refer to both full trees and subtrees, unless otherwise specified.

[2] `A > B` means A governs B; `A < B` means A is governed by B; dependency labels follow the operator (e.g., `A >obj B`); and parentheses (e.g., `A > (B > C)`) mark attachment priority. Letters like A and B stand for tokens, which can be constrained by form, lemma, UPOS, etc. The underscore (_) represents any token.

- **Head constraints** (--head) limit tree extraction to structures rooted in tokens matching a specified property, such as upos=NOUN or lemma=want. This is useful for focusing on specific construction types (e.g., noun-headed phrases, as in Table 2), or for studying lexicogrammatical behavior of individual words.

- **Label constraints** combine two parameters (--allowed_labels and --ignored_labels) to restrict which relations can appear in a tree. Users can specify a whitelist of allowed relations (e.g., nsubj|obj|iobj), or indicate relations to ignore as irrelevant (e.g., punct), without discarding the tree itself.

- **Custom queries** (--query) provide fine-grained control by allowing users to specify an exact dependency pattern to match.[3] Crucially, STARK applies all other representation settings (see Section 2.2) when generating the output, enabling hybrid workflows that combine top-down targeting with bottom-up extraction—e.g., listing all lexical realizations or surface order permutations of a pattern (as in Table 4, for example).

These flexible and combinable filters give users precise control over the granularity and scope of extraction, making STARK adaptable to a wide range of research goals.

2.4 Optional Processing Mode

By default, STARK extracts full subtrees rooted at each token—i.e., the head and all direct/indirect dependents—producing syntactically coherent units. The --complete parameter can be adjusted to instead extract all connected subtrees anchored at a token, including partial or nested fragments. While this mode can reveal finer combinatorial detail, it is computationally more demanding and best suited for small datasets or targeted analyses.

3 Statistical Analysis

In addition to extracting and representing syntactic structures, STARK provides a range of statistical measures that support quantitative corpus-based syntactic analysis. These include basic frequency counts, association scores, and keyness comparisons, all computed based on the extracted trees. In this section, we illustrate each type of analysis on different corpora and configurations to highlight STARK's flexibility.[4]

3.1 Frequency

By default, STARK outputs absolute and relative frequency counts for each extracted tree. Relative frequencies are normalized per million tokens, enabling comparison across corpora of different sizes. This information is useful for identifying both dominant constructions and rare syntactic patterns (including annotation mistakes), providing insight into the overall distribution of specific structures in a corpus. For example, Table 2 in Appendix A lists the ten most frequent noun-headed trees in the English GUM UD Treebank (Zeldes, 2017), revealing the most common types of nominal phrase patterns in the language that can inform descriptive grammar work and usage-based models, or serve as a basis for comparative studies.

3.2 Association

In addition to frequency, STARK optionally computes several statistical association measures via the --association_measures parameter. These quantify the strength of co-occurrence between nodes within a tree (Evert, 2009) and include mutual information (MI), MI^3, Dice, logDice, t-score, and log-likelihood (LL). The scores are particularly useful for identifying collocationally strong structures, especially in lexicalized output. This is illustrated in Table 3 in Appendix A, which lists the top ten noun phrases of size 3 or more in the French GSD UD treebank (Guillaume et al., 2019) ranked by logDice, revealing a range of nominal multi-word expressions.

3.3 Keyness

STARK supports keyness analysis via the --compare parameter, which compares extracted trees against a reference corpus. It calculates the relative frequency of each tree in both corpora and computes several keyness scores (Gabrielatos, 2018), including log-likelihood (LL), BIC, log ratio, odds ratio, and %DIFF. These help detect struc-

[3] Currently, queries are written in dep_search (e.g. 'upos=VERB >nsubj _ >obj _' for verb-subject-object trees retrieved in Table 4), but support for other query languages like Grew (Guillaume, 2021) or Semgrex (Bauer et al., 2023) could be added in future.

[4] Due to space constraints, more information on the specific measures is available in the cited literature and at: https://github.com/clarinsi/STARK/blob/master/statistics.md.

tures that are disproportionately frequent or underrepresented in one corpus relative to another, making this feature particularly useful for comparing syntactic or lexical behavior across genres, domains, or languages.

Table 4 in Appendix A illustrates this by comparing subject–verb–object (SVO) patterns in the spoken (SST) and written (SSJ) Slovenian UD treebanks (Dobrovoljc et al., 2017; Dobrovoljc and Nivre, 2016), highlighting constructions that are more or less prominent in speech in comparison to writing.

4 Example Retrieval and Visualisation

STARK offers optional output enhancements to support qualitative analysis and visualization. Users can retrieve a sample sentence per tree (`--example`), with marked nodes, or add node-level (`--node_info`) and head-level (`--head_info`) details for further analysis.

STARK also supports integration with online treebank browsing services. If the input is an official UD treebank (i.e., follows the standard naming convention), enabling the `--grew_match` option generates clickable links to the corresponding patterns in the Grew-match interface (Guillaume, 2021).[5] These links let users explore all instances of a given tree in context within the latest UD release and leverage additional Grew-match functionalities.

For compatibility with legacy tools, the `--depsearch` option outputs trees in the `dep_search` syntax used by earlier platforms such as SETS (Luotolahti et al., 2015), dep_search (Luotolahti et al., 2017) or Drevesnik (Štravs and Dobrovoljc, 2024).[6]

5 Scalability

STARK has been tested on all official UD treebanks and can handle corpora of various sizes and annotation styles, including multi-root sentences and non-standard labels. Output volume can be managed via frequency thresholds (`--frequency_threshold`) or by capping the number of output trees (`--max_lines`), making it easy to scale STARK to large datasets while maintaining interpretability.

Several advanced settings have also been introduced to further improve performance and scalability. These include multi-core support (`--cpu_cores`), internal caching for repeated experiments (`--internal_saves`), and chunked processing of directory-based corpora (`--continuation_processing`). Users can also select between two extraction modes: the default `greedy_counter`, optimized for bottom-up tree extraction, and the `query_counter`, which performs better when used with specific target patterns.

6 Availability and Online Demo

STARK is freely available as an open-source tool under the Apache 2.0 license.[7] In addition to the command-line interface, the tool is also released as a Python library via PyPI (`pip install stark-trees`),[8] enabling seamless integration into custom scripts and larger NLP workflows. Comprehensive documentation is available through the GitHub and PyPI repositories where users can find detailed explanations of all parameters, usage examples, and configuration tips.

To further support accessibility, STARK is also available via an interactive online demo.[9] The web interface covers all core functionalities of the tool, allowing users to select a treebank, configure extraction settings (see Section 2), and explore the output in an interactive table view. Unlike the command-line version, the online interface also provides visualisations for one or multiple example instances of the tree.

As such, the online demo is particularly useful for exploratory browsing, classroom use, and first-time users unfamiliar with the command-line interface. Screenshots of both the settings panel and the output view are shown in Figures 2 and 3 in Appendix B.

7 Conclusion

We introduced STARK, a versatile toolkit for bottom-up syntactic analysis of dependency-parsed corpora. By extracting, ranking, and comparing syntactic (sub)trees, it enables exploratory and data-driven research without requiring predefined queries. The tool supports a wide range of configurations and outputs, and is available as a command-line tool, Python library, and online demo.

Its practical value has already been demonstrated through early adoption in a range of re-

[5] https://universal.grew.fr

[6] https://orodja.cjvt.si/drevesnik/

[7] https://github.com/clarinsi/STARK

[8] https://pypi.org/project/stark-trees/

[9] https://orodja.cjvt.si/stark/

search contexts, from integration into tools such as the DELTA diversity pipeline (Estève and Dobrovoljc, 2025) and ComparaTree treebank comparison tool (Terčon and Dobrovoljc, 2025), to studies on syntactic profiling of spoken data (Hüll and Dobrovoljc, 2025; Dobrovoljc, 2025), learner essays (Munda and Holdt, 2025), and parallel multilingual corpora (Čibej, 2025).

Acknowledgments

This work was financially supported by the Slovenian Research and Innovation Agency through the research projects Treebank-Driven Approach to the Study of Spoken Slovenian (Z6-4617), Large Language Models for Digital Humanities (GC-0002), and the research program Language Resources and Technologies for Slovene (P6-0411). Special thanks go to Marko Robnik Šikonja for his early contributions to the project, and to CLARIN.SI for additional support through the 2019 and 2024 Resource and Service Development grants. We also thank Tina Munda, Nives Hüll, Luka Terčon, and Louis Estève for their help with testing and feedback. Generative AI tools were employed to support language editing. The authors retain full responsibility for the content, analysis, and interpretation presented in this work.

References

John Bauer, Chloé Kiddon, Eric Yeh, Alex Shan, and Christopher D. Manning. 2023. Semgrex and ssurgeon, searching and manipulating dependency graphs. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 67–73, Washington, D.C. Association for Computational Linguistics.

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

Kaja Dobrovoljc. 2025. Counting trees: A treebank-driven exploration of syntactic variation in speech and writing across languages. *Preprint*, arXiv:2505.22774.

Kaja Dobrovoljc, Tomaž Erjavec, and Simon Krek. 2017. The Universal Dependencies treebank for Slovenian. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, pages 33–38, Valencia, Spain. Association for Computational Linguistics.

Kaja Dobrovoljc and Joakim Nivre. 2016. The Universal Dependencies treebank of spoken Slovenian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1566–1573, Portorož, Slovenia. European Language Resources Association (ELRA).

Louis Estève and Kaja Dobrovoljc. 2025. A new pipeline for measuring diversity across various linguistic levels. Abstract accepted at the UNIDIVE 3rd General Meeting in Budapest.

Stefan Evert. 2009. *58. Corpora and collocations*, pages 1212–1248. De Gruyter Mouton, Berlin, New York.

Ramon Ferrer-i Cancho, Carlos Gómez-Rodríguez, Juan Luis Esteban, and Lluís Alemany-Puig. 2022. Optimality of syntactic dependency distances. *Phys. Rev. E*, 105:014308.

Costas Gabrielatos. 2018. *Keyness Analysis: nature, metrics and techniques*, pages 225–258. Routledge, United Kingdom.

Bruno Guillaume. 2021. Graph matching and graph rewriting: GREW tools for corpus exploration, maintenance and conversion. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–175, Online. Association for Computational Linguistics.

Bruno Guillaume, Marie-Catherine de Marneffe, and Guy Perrier. 2019. Conversion et améliorations de corpus du français annotés en Universal Dependencies. *Revue TAL : traitement automatique des langues*, 60(2):71–95.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Nives Hüll and Kaja Dobrovoljc. 2025. Word order variation in spoken and written corpora: A cross-linguistic study of svo and alternative orders. In *Proceedings of the SyntaxFest 2025*. To appear.

Natalia Levshina. 2022. Corpus-based typology: applications, challenges and some solutions. *Linguistic Typology*, 26(1):129–160.

Yuri Lin, Jean-Baptiste Michel, Erez Lieberman Aiden, Jon Orwant, Will Brockman, and Slav Petrov. 2012. Syntactic annotations for the google books ngram corpus. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, page 169–174, USA. Association for Computational Linguistics.

Juhani Luotolahti, Jenna Kanerva, and Filip Ginter. 2017. Dep_search: Efficient search tool for large dependency parsebanks. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 255–258, Gothenburg, Sweden. Association for Computational Linguistics.

Juhani Luotolahti, Jenna Kanerva, Sampo Pyysalo, and Filip Ginter. 2015. Sets: Scalable and efficient tree search in dependency graphs. In *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Demonstrations*, pages 51–55.

Tina Munda and Špela Arhar Holdt. 2025. First insights into the syntax of slovene student writing: A statistical analysis of šolar 3.0 vs. učbeniki 1.0. In *Proceedings of the SyntaxFest 2025*. In print.

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

Jan Štepánek and Petr Pajas. 2010. Querying diverse treebanks in a uniform way. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 1828–1835.

Miha Štravs and Kaja Dobrovoljc. 2024. Service for querying dependency treebanks drevesnik 1.1. Slovenian language resource repository CLARIN.SI.

Luka Terčon and Kaja Dobrovoljc. 2025. Comparatree: A multi-level comparative treebank analysis tool. In *Proceedings of the SyntaxFest 2025*. To appear.

Yaqin Wang and Haitao Liu. 2017. The effects of genre on dependency distance and dependency direction. *Language Sciences*, 59:135–147.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

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

Jaka Čibej. 2025. A computational method for analyzing syntactic profiles: The case of the elexis-wsd parallel sense-annotated corpus. In *Proceedings of the SyntaxFest 2025*. To appear.

A Example STARK Outputs

Rank	Tree	Freq.	Example
1	NOUN	4436	*They're bringing* ***drugs***.
2	DET <det NOUN	2331	*Plant* ***the cuttings***.
3	ADP <case DET <det NOUN	1874	*Remove* ***from the oven***.
4	ADP <case NOUN	1815	*Prepare* ***for impact***.
5	CCONJ <cc NOUN	809	*Distinguish concepts* ***and prototypes***.
6	PRON <nmod:poss NOUN	803	***My house*** *was empty and cold.*
7	ADJ <amod NOUN	735	*We have* ***hard work*** *ahead.*
8	ADP <case ADJ <amod NOUN	602	*They show it* ***in many ways***.
9	DET <det ADJ <amod NOUN	576	*Yeah, or turn* ***a deaf ear***.
10	ADP <case PRON <nmod:poss NOUN	569	*He was recognized for some* ***of his books***.

Table 2: Top 10 nominal phrase structures in the English GUM UD Treebank sorted by frequency. STARK settings used: `node_type = upos, labeled = yes, label_subtypes = yes, fixed = yes, size = 1-10000, head = upos=NOUN, complete = yes`.

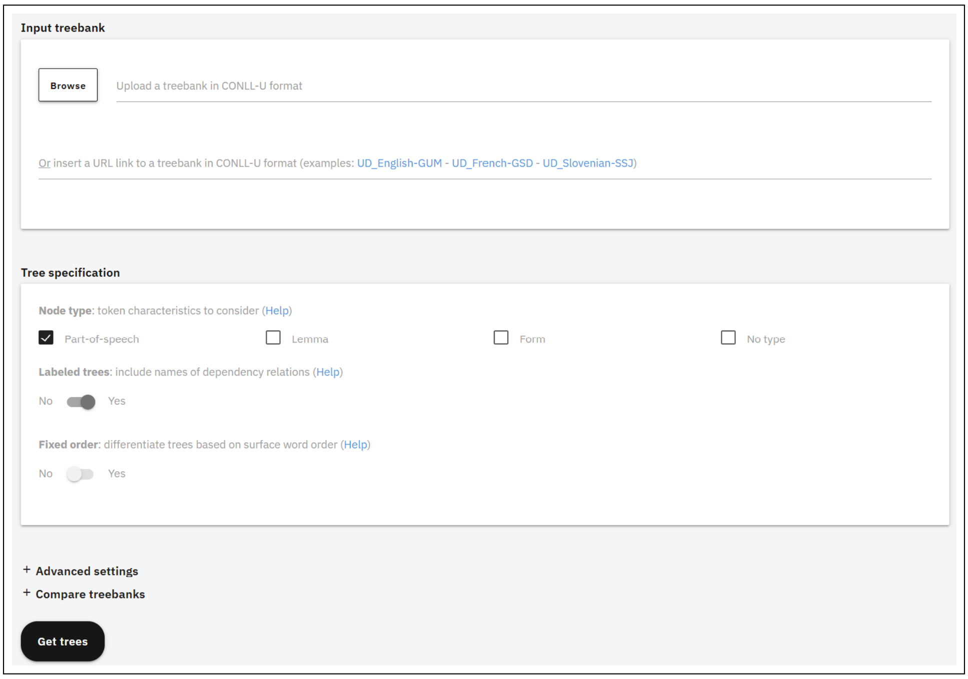
Rank	Tree	Freq.	No. of nodes	logDice
1	qualité > / < prix	5	3	9.35
2	pour < la < première < fois	35	4	7.50
3	une < nouvelle < fois	10	3	6.97
4	de < le < monde	92	3	6.78
5	pour < sa < part	7	3	6.70
6	par < la < suite	24	3	6.58
7	sur < des < prises	5	3	6.19
8	d' < araignées > aranéomorphes	7	3	6.12
9	l' < année > suivante	9	3	6.02
10	de < la < ville	48	3	6.00

Table 3: Top 10 nominal multi-word expressions in the French GSD UD Treebank ranked by logDice association measure. STARK settings used: `node_type = form, labeled = no, fixed = yes, size = 3-10, head = upos=NOUN, complete = yes, association_measures = yes, frequency_threshold = 5`.

Rank	Tree	RF in SST	RF in SSJ	OR	Example
1	_ <nsubj _ <obj _	1768.4	1639.9	1.08	*če naši* ***možje to naredijo***.
2	_ <obj _ <nsubj _	1321.2	1299.2	1.02	***tega nihče*** *ni* ***razumel*** *dolgo.*
3	_ >nsubj _ >obj _	396.4	408.1	0.97	*pa* ***imam jaz*** *tudi svoje* ***obveznosti***.
4	_ >obj _ >nsubj _	203.3	325.7	0.62	***zanimala*** *sta* ***vas novinarstvo*** *in filozofija*
5	_ <obj _ >nsubj _	1473.7	3159.9	0.47	***kaj izraža*** *ta* ***glagol*** *?*
6	_ <nsubj _ >obj _	3323.4	8225.5	0.40	*katera* ***črta razpolavlja kot*** *?*

Table 4: SVO patterns in the spoken Slovenian SST UD Treebank ranked by Odds Ratio (OR) keyness measure, when compared to the written SSJ UD Treebank. RF = relative frequency. STARK settings used: `labeled = yes, fixed = yes, query = upos=VERB >nsubj _ >obj _, complete = no, compare = sl_ssj-ud.conllu`.

B STARK Web Interface



Input treebank

Browse | Upload a treebank in CONLL-U format

Or insert a URL link to a treebank in CONLL-U format (examples: UD_English-GUM - UD_French-GSD - UD_Slovenian-SSJ)

Tree specification

Node type: token characteristics to consider (Help)

☑ Part-of-speech ☐ Lemma ☐ Form ☐ No type

Labeled trees: include names of dependency relations (Help)

No ● Yes

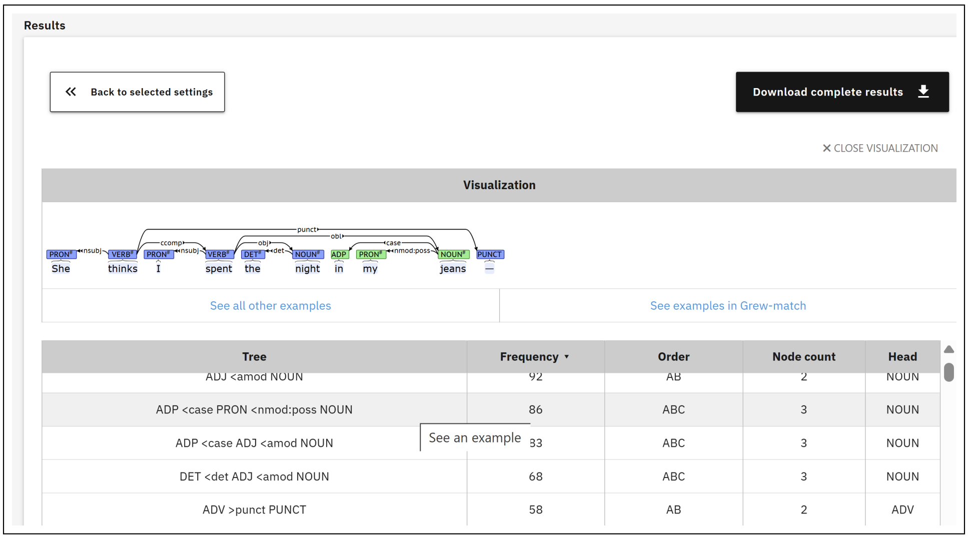
Fixed order: differentiate trees based on surface word order (Help)

● No Yes

+ Advanced settings
+ Compare treebanks

Get trees

Figure 2: Screenshot of the STARK online demo interface, showing the interactive settings selection, from basic tree specification to advanced tree filtering and treebank comparison options.



Results

« Back to selected settings

Download complete results

✕ CLOSE VISUALIZATION

Visualization

See all other examples | See examples in Grew-match

Tree	Frequency ▾	Order	Node count	Head
ADJ <amod NOUN	92	AB	2	NOUN
ADP <case PRON <nmod:poss NOUN	86	ABC	3	NOUN
ADP <case ADJ <amod NOUN	33	ABC	3	NOUN
DET <det ADJ <amod NOUN	68	ABC	3	NOUN
ADV >punct PUNCT	58	AB	2	ADV

See an example

Figure 3: Screenshot of the STARK online demo interface, showing the interactive results table, an example tree visualisation, and links to explore all matched examples in both the demo and Grew-match.

«Are you Afraid of Ghosts?»
A Proposal for Busting Predicate Ellipsis in Universal Dependencies

Claudia Corbetta
Università di Bergamo-Pavia
via Salvecchio 19,
24129 Bergamo, Italy.
claudia.corbetta@unibg.it

Federica Iurescia and **Marco Passarotti**
Università Cattolica del Sacro Cuore
CIRCSE Research Centre
Largo Gemelli, 1, 20123 Milan, Italy
{federica.iurescia,marco.passarotti}@unicatt.it

Abstract

This paper addresses the representation of ellipsis in dependency syntax, proposing both a theoretical and a practical workflow for its analysis and annotation in treebanks, following the state-of-the-art Universal Dependencies framework. We discuss the challenges of annotating ellipsis, with a focus on predicate ellipsis and its representation in dependency treebanks, and emphasize the importance of accounting for such phenomena for syntactic analysis and machine learning applications. We present a case study based on the Italian-Old treebank, demonstrating the applicability of the proposed workflows and invite the community to participate in this initiative with their own languages.[1]

1 Introduction

A widely acknowledged principle in science is that the manner in which we choose to represent reality significantly shapes the nature of the material we aim to structure and interpret (Kuhn, 1962). This is particularly true in the analysis of ellipsis, a syntactic phenomenon that represents the omission of linguistic material in a sentence (Merchant, 1999), where the choice of the model to represent and encode (missing) linguistic information has a crucial impact on both its study and interpretation.

Studying a phenomenon that represents, by its nature, an absence is a challenging task. Merchant (Merchant, 2018, p.25) draws a compelling comparison of ellipsis with a black hole, stating that “detecting and arguing for such “missing” structures is analogous to searching for and determining the properties of a black hole: one can tell it’s there only by its effects on surrounding material”. Due to this inherently elusive nature, since ellipsis is, by definition, silent in the data, an additional challenge arises in representing it within syntactically annotated corpora (treebanks), which are designed, among other purposes, to support the representation and queryability of syntactic phenomena.

In the present paper, we address the representation of ellipsis in dependency syntax by providing both a theoretical and a practical workflow for analyzing and representing it in treebanks, in accordance with the state-of-the-art dependency framework, Universal Dependencies (De Marneffe et al., 2021).

The paper is structured as follows: in Section 2, we provide a definition of ellipsis and outline key concepts. Section 3 offers an overview of ellipsis within syntactic theory, with particular attention on dependency frameworks. It also presents how ellipsis is addressed in various dependency treebanks, with a specific focus on Universal Dependencies in Subsection 3.1. In Section 4, we discuss the importance of annotating ellipsis. Sections 5 and 6 introduce the theoretical and practical workflows, respectively, along with the challenges they entail. Finally, Section 7 presents a case-study based on the Italian-Old treebank, and Section 8 concludes the paper.

2 What is Ellipsis?

Ellipsis has been defined as an asymmetry between meaning and form in an expression (Van Craenenbroeck and Temmerman, 2018).

In this Section, we will address some key concepts about ellipsis, in order to provide terminology and insights of the phenomenon. When speaking of ellipsis, we should consider the following aspects:

- **elided site**: the elided site refers to the position of the ellipsis, namely it represents the gap in the sentence where the linguistic mate-

[1] This paper is the result of a collaboration among all three authors. In accordance with the requirements of the Italian academic attribution system, Claudia Corbetta is responsible for the entire paper. Claudia Corbetta, Federica Iurescia, and Marco Passarotti are specifically responsible for Section 6.

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 52–63
August 28-29, 2025 ©2025 Association for Computational Linguistics

rial is omitted. It is usually represented with "___" in the sentence, whereas in the syntactic structure it can be represented with an empty node.

- **remnants**: the remnants are the "survivors" in an elliptical sentence (Ortega-Santos et al., 2014, p. 55). This term refers to the linguistic material that is not elided in a clause that presents an ellipsis.
- **elided material**: it refers to the linguistic material that is omitted, and therefore that undergoes to ellipsis. In the literature, it is usually represented inside square brackets [].[2]
- **antecedent**: the antecedent is the linguistic material that leads the speaker/reader to understand and correctly process the ellipsis. It can be explicitly expressed in the sentence, or in the text, but it can also be inferred from world knowledge, i.e., not explicitly stated, but still recoverable by the listener/reader. In the literature, the term antecedent is always used in a broader way, not obligatory referring to an element that precedes the ellipsis site. In fact, the antecedent can also follow the ellipsis site, thus making more appropriate the term "postcedent" (McShane, 2005, p. 14).[3]
- **identity condition**: the identity condition has been debated by several scholars. It refers to the identity between the antecedent and the elided material. We will show in Section 6, that it is not always the case.

In Example 1, we provide an instance of ellipsis, highlighting all the aspects shown above:

Example 1
"I wish all happy holidays, and moreso,
___ peace on earth."[4]

The ellipsis site is marked with the "___". The remnants are "and", "moreso", "peace on earth". The antecedent that lead as to solve the ellipsis is composed by the subject "I", the verb "wish", the beneficiary "all", and, in this case, the elided material respects the identity condition, being a copy of the antecedent.[5] Therefore, the sentence with the elided material expressed will be as follows:

"I wish all happy holidays, and moreso,
[I wish all] peace on earth."

3 Ellipsis in Syntax and in Treebanks

From a **theoretical syntactic perspective**, studies on ellipsis have been conducted mainly within constituency frameworks (Ross, 1969; Merchant, 2001; Kennedy, 2003), which significantly outnumber those grounded in dependency syntax. Constituency-based analysis of ellipsis tends to provide a broad classification of ellipsis types,[6] primarily grounded in the notion of constituent movement within the syntactic structure.

However, within the dependency framework, ellipsis has not been extensively analyzed, positioning it as a relatively underexplored syntactic phenomenon. Among the main studies on ellipsis from a dependency perspective, Osbourne (Osborne, 2019) offers a key contribution, namely the identification of the *catena*, a syntactic unit different from the constituent. His analysis and classification of ellipsis build and provide justification of licensing different type of ellipsis. Ellipsis has also been defined as "unrealized words" by Hudson (Hudson, 2010), referring to covert words that lack pronunciation or spelling, with their unique distinction from overt ones being their inaudibility.

Regarding the representation of **ellipsis in treebanks**, approaches vary depending on the formalism adopted. In constituency-based treebanks, such as the Penn Treebank (PTB) (Marcus et al., 1993) and the BulTreeBank (Osenova and Simov, 2003), which follows the Head-Driven Phrase Structure Grammar (HPSG) formalism (Pollard and Sag, 1994), ellipses are explicitly annotated through the use of empty nodes.

Conversely, in dependency treebanks, the representation of ellipsis poses a greater challenge. This

[2] We will not address the question of whether the elided material exists at a cognitive level. For a general overview of how psycholinguistics addresses questions concerning the representation of sentences involving ellipsis, see Phillips and Parker (2014). However, it is clear that, in order to analyze ellipsis, it must be made explicit.

[3] We will align with the literature (McShane, 2005, p.14) and use the term antecedent as a macro-term that is not connected with its position with respect to the ellipsis site.

[4] This sentence (ENG_20041111_173500-0051) is taken from the UD_English-EWT. See: `https://github.com/UniversalDependencies/UD_English-EWT`.

[5] Within the UD framework, such an example is treated as ellipsis and annotated with the `orphan` relation (see Section 3.1 and 3.2 for further discussion of `orphan`). The presence of the adverb *moreso* provides clear evidence of a missing verb, which would constitute its syntactic head.

[6] For an overview on ellipsis classification, see Van Craenenbroeck and Temmerman (2018).

is primarily due to the principle that, in dependency-based annotation, the number of nodes in the tree corresponds exactly to the number of tokens in the sentence, thereby precluding the use of empty nodes. For instance, the Prague Dependency Treebank (PDT) (Hajič et al., 2017) addresses ellipsis explicitly through a dedicated attribute and requires its reconstruction at other annotation layers.[7] In the following Subsections 3.1 and 3.2, we address in detail the formalism adopted by the state-of-the-art dependency framework, namely Universal Dependencies.

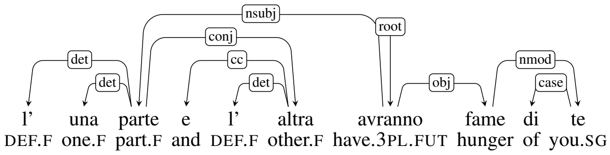
3.1 Ellipsis in Basic Universal Dependencies

When it comes to syntactic dependency resources, the state-of-the-art framework is Universal Dependencies (henceforth UD) (De Marneffe et al., 2021), which currently includes 319 treebanks covering 179 languages.[8]

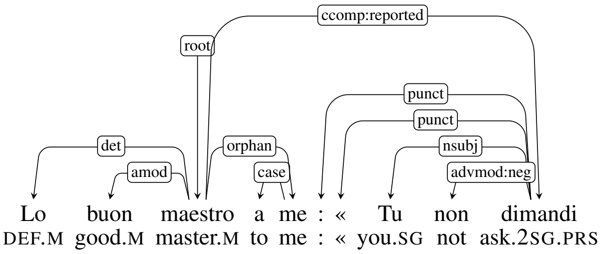
UD provides two levels of syntactic annotation: a basic layer and an enhanced one, called Enhanced Dependencies. The basic annotation includes only overt words (i.e., non-null nodes) and therefore does not allow for empty nodes.[9] As a result, ellipsis is not explicitly represented in the basic layer. Instead, when annotating elliptical structures, the UD guidelines recommend either adopting a promotion strategy,[10] if the syntactic structure remains grammatically well-formed, or using the dependency relation `orphan` when promotion would result in an ungrammatical configuration.

Example 2 illustrates the use of the promotion strategy in an Old Italian sentence from the Italian-Old treebank, whereas Example 3 provides an instance of the `orphan` relation:

Example 2 - *Inf.* XV, vv. 71-72
l'una parte e l'altra avranno fame/ di te
"one party and the other will be hungry/ for you"[11]



Example 3 - *Inf.* IV, v. 31
Lo buon maestro a me:/ «Tu non dimandi (...)
"To me the Master good:/ "Thou dost not ask (...)"



In Example 2, the node *altra* "other" is promoted to the head of the conjunct `conj`, which involves a case of nominal ellipsis, with the noun *parte*, "part" being elided. By contrast, in Example 3, applying the promotion strategy would have resulted in an ungrammatical syntactic structure. Therefore, the noun *maestro* "master" is promoted to the root of the sentence, and the prepositional phrase *a me* "to me", which functions as an oblique, is attached to the nominal root using the `orphan` relation.

As the evidence shows, neither of the proposed solutions directly addresses the annotation of ellipsis. Instead, they offer workarounds that attempt to accommodate elliptical constructions within the annotation scheme and without the construction of empty nodes.

3.2 Ellipsis in Enhanced Universal Dependencies

However, empty nodes are permitted in **Enhanced Dependencies** (henceforth EUD). EUD is an extension of basic UD, designed to make «some of the implicit relations between words more explicit».[12] Among these explicit relations, ellipsis is also addressed. Specifically, as outlined in the guidelines,[13] predicate ellipsis permits the insertion of an empty node, allowing for the restoration of syntactic relations that would otherwise be lost in the annotation. Concerning the annotation of the empty node, information on form, lemma, and

[7] See Mikulová (2014) for further discussion of ellipsis in the PDT.

[8] These numbers refer to version 2.16. See: https://universaldependencies.org.

[9] In this work, we will use the terms "empty nodes" rather than "null nodes". Although these terms are often used interchangeably in the UD guidelines, in linguistic literature null node is frequently associated with valency-related phenomena, such as null subjects or objects. Accordingly, the term "empty node" is adopted here, as it more accurately reflects the syntactic nature of the structure.

[10] See the guidelines: https://universaldependencies.org/u/overview/specific-syntax.html#ellipsis.

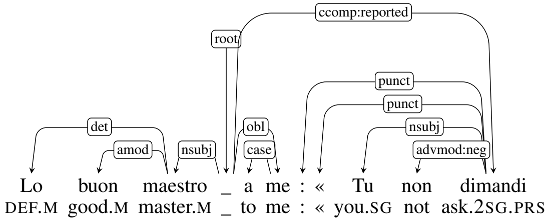
[11] The English translations of the examples from the Comedy are by Allen Mandelbaum, available at: https://digitaldante.columbia.edu/dante/divine-comedy/.

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

[13] https://universaldependencies.org/u/overview/enhanced-syntax.html#ellipsis.

UPOS, XPOS and features is optionally "copied from the overt occurrence of the predicate".[14]

We present the enhanced annotation of the sentence in Example 3.[15]


ccomp:reported
root
punct
punct
det
obl
nsubj
amod
nsubj
case
advmod:neg
Lo buon maestro _ a me : « Tu non dimandi
DEF.M good.M master.M _ to me : « you.SG not ask.2SG.PRS

The graph represents the empty node, visually indicated by the underscore "_" in the sentence. Meanwhile, the noun *maestro* "master" is now attached to the root with the subject relation (nsubj), and the prepositional phrase and the complement clause depend on the root as an oblique (obl) and a clausal complement (ccomp:reported), respectively.

Even though EUD is crucial for analyzing phenomena like ellipsis, unfortunately, there are few treebanks that provide enhanced annotation,[16] resulting in a scarcity of gold-standard data for the phenomena specifically addressed by EUD.

In the following Sections 4, 5, and 6, we will introduce and justify the need for gold-annotated data for ellipsis, and propose possible methods to address this gap. More specifically, in Section 4, we will describe the challenges in automatically enhancing ellipsis. In Section 5, given the unexplored nature of the topic from a corpus-based perspective, we will suggest a general approach to investigate ellipsis in this context, also discussing the challenges encountered in doing so. In Section 6, we will specify the approach within the UD framework to provide a method for enriching the annotation of this valuable dataset.

4 Why do we Need Ellipsis?

As mentioned in Section 3.1, only a small amount of enhanced treebanks is available. Providing an enhanced version is not mandatory in UD, and in most cases, the effort required to develop and maintain a basic treebank is already substantial enough to relegate the manual creation of an enhanced layer to a lower priority. Additionally, the task of automatically enhancing a basic UD treebank is particularly challenging, especially when it comes to generating empty nodes to account for ellipsis (Simi et al., 2018; Droganova et al., 2018).

Moreover, attempts to automatically parse ellipsis in dependency treebanks using large language models (LLMs) have not yielded satisfactory results (Ćavar et al., 2024a,b), which has led to the development of corpora specifically focused on ellipsis.[17]

While it is clear that, at the current state of the art, a fully automatic reconstruction of ellipsis is not yet feasible, it is still possible to support the human annotation process by designing workflows that facilitate the manual annotation of ellipsis.

Before describing the procedures that assist manual annotation, in Section 5, we will briefly outline the general steps that are essential for dealing with ellipsis. Specifically, in this exploratory study, we focus on predicate ellipsis.[18]

5 "Ghostbusting" Ellipsis: A Theoretical Approach

Before moving on to a practical proposal for how to address ellipsis in UD (see Section 6), we will first outline the general workflow required when dealing with ellipsis, namely with something that is absent from the data. We divide this workflow into three steps: detection, identification, and reconstruction. It is important to clarify that the proposed workflow - along with the order in which the steps are presented - is intended as a methodological proposal for handling ellipsis in annotated data. In other words, it does not address how ellipsis is processed in the mind of the reader or speaker - particularly with respect to the reconstruction phase - which is the subject of specific psycholinguistic studies (Hofmeister, 2007), but lies beyond the scope of the present work. Each step is described in detail below:

- **detection**: this step refers to the detection of the ellipsis site, that is, recognizing the presence of an ellipsis in the text. While for certain linguistic phenomena the detection task is relatively straightforward, this is not the case for ellipsis, which involves an omission. In

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

[15] In Appendix A, we report the full tree with both the basic and the enhanced annotation (See Example 3.1).

[16] To date, only 19 treebanks include all types of enhancements. Moreover, EUD treebanks are difficult to quantify, as they vary in the specific enhancements they incorporate.

[17] An example is the Hoosier Corpora: https://github.com/dcavar/hoosierellipsiscorpus.

[18] We leave for future works the management of other types of ellipsis, such as nominal ellipsis.

fact, determining whether something is missing requires careful consideration, in order to avoid overgeneralizing ellipsis and identifying ellipsis sites where there are none. Particular attention should be paid to drawing a clear boundary between ellipsis and coordination. This issue will be further discussed in Section 6.

- **identification**: once the presence of an ellipsis has been detected, the next step is to concretize it by retrieving its antecedent, which may be found in the same sentence of the ellipsis site, elsewhere in the text, or inferred from world knowledge. This task is closely related to coreference and anaphora resolution,[19] as it involves identifying (or understanding) the linguistic material that supports the interpretation of the ellipsis - that is, its antecedent.

- **reconstruction**: the final step in dealing with ellipsis is the reconstruction of the elided material (or, in terms of syntactic structures, the empty nodes), and, if possible, the annotation of its linguistic information (Section 7.2). It is important to emphasize that this process is conceived as a practical task for ellipsis retrieval and annotation, in line with the principle that the more information we have in annotation, the better it is.

However, each of these steps raises questions and presents challenges that deserve to be addressed and discussed. We will list them in their respective order, presenting the problems in forms of questions:

- **where should the ellipsis site be placed?** The first challenge in the detection process involves the position of the ellipsis site. Determining the position of the ellipsis site can become problematic, especially when languages with a relatively free word order are concerned. We will discuss a possible solution to this issue in Section 7.

- **how to identify the antecedent?** As mentioned in Section 2, the antecedent is not always present in the sentence where the ellipsis occurs. Sometimes, it may not even be present in the text at all. While recognizing an antecedent within the text is a challenging task (especially for a machine), identifying it from world knowledge is even more difficult, raising the question of whether and how it is possible to circumscribe it.

- **what, if anything, should we reconstruct?** Since the aim is to make ellipsis explicit in order to analyze and recognize it, it is evident that reconstruction plays a crucial role in the task. However, it also presents significant challenges. The main risk is creating a cemetery of empty nodes, where the reconstruction of missing elements is exceedingly arbitrary. Therefore, it is essential to carefully consider the extent of reconstruction of the empty node, both in terms of how much we should reconstruct (i.e., where the antecedent ends) and what information about the reconstructed node should be reported in the empty node.

We will suggest possible solutions to some of these issues in the following Sections.

6 "Ghostbusting" Ellipsis: A Practical Workflow to Deal with Ellipsis in (E)UD

Building upon the considerations in Section 5, we provide in this Section a practical workflow for addressing ellipsis in (E)UD. This Section is structured as follows: Subsection 6.1 outlines a method for querying ellipsis in basic treebanks, while Subsection 6.2 presents a new proposal for annotating ellipsis in EUD.

6.1 Detection: How to Find Something that Is Not There

The initial step in implementing ellipsis in an enhanced annotated treebank is the **retrieval** of ellipsis instances from the data. Even though, given the current state-of-the-art NLP tools, detecting ellipsis in treebanks remains primarily a task that must be carried out manually, it is still possible to develop methods that facilitate and accelerate the detection and extraction process.

This method was developed and tested on the Italian-Old treebank (see Section 7), which documents an Old Italian poem. However, the proposed method is generalizable and can be applied to other languages as well, with possible minor adjustments (See 6.2, footnote 25).

The retrieval of ellipsis in basic UD treebanks involves two steps.

[19] See, in this regard, the analysis by Hankamer and Sag (1976) on ellipsis as a form of surface anaphora.

- The first, straightforward step is to search for the **orphan** dependency relation, as this label is specifically employed in cases of (predicate) ellipsis.

- The second step focuses on identifying instances annotated using the **promotion** strategy (see Subsection 3.1),[20] where the dependent remnant of the ellipsis is promoted and assigned the dependency relation of the elided node. This second strategy relies on detecting a **mismatch** between the morphological annotation, namely the Part-of-Speech (henceforth PoS) of the remnant and the syntactic function it inherits, one that is not prototypical for its PoS. For instance, in the case of nominal ellipsis involving the promotion of an adjective, ellipsis can be identified by querying for adjectives (ADJ) that bear dependency relations typically associated with nouns (NOUN), such as subject (`nsubj`) or object (`obj`).

While the first step offers a direct and effective method for retrieving some instances of (predicate) ellipsis,[21] the second strategy aims to retrieve additional (and not explicitly annotated) cases that *may* involve ellipsis. Naturally, a manual inspection remains necessary in order to filter and evaluate genuine instances of ellipsis, distinguishing them from possible false positives.

Similar queries can be performed using available tools such as Udapi (Popel et al., 2017), Grew-Match (Bonfante et al., 2018), or ArboratorGrew (Guibon et al., 2020), among others.

In Section 7, we will provide a practical example of retrieving predicate ellipsis in the Italian-Old treebank, applying both of these strategies.

6.2 Identification and Reconstruction: Proposal for Common Annotation for Predicate Ellipsis

Once ellipsis has been detected, the next step is to mark it as such. In EUD, this involves creating an empty node and restoring the dependency relations that were lost in basic UD (see Subsection 3.1). However, as discussed in Section 4, the challenges in handling ellipsis and empty nodes have led to the lack of in-depth enhanced guidelines, specifically for ellipsis annotation.

Accordingly, this section aims to offer proposals to address proper and consistent ellipsis annotation in the UD framework, building upon the theoretical considerations outlined and addressing the issues raised in in Section 5. Specifically, we will address each issue raised in Section 5, and provide a possible solution:

- **where should the ellipsis site be placed?** we pursue an approach based on parallelism,[22] which involves mirroring the order of the phrases present in the antecedent. In cases where the antecedent is not present, we suggest following the canonical word order of the sentence.

  For instance, in the Example 1 “I wish all happy holidays, and moreso, peace on earth”, the ellipsis site mirrors the order of the sentence with the antecedent: it precedes the object (“peace on earth”) (“I wish all happy holidays, and moreso, [*empty node*] peace on earth”). More complex cases will be discussed in Section 7;

- **how to identify the antecedent?** As mentioned in Section 2, the identification of the antecedent is a crucial step in the task of processing ellipsis. Since this work aims to provide an annotation of ellipsis, tracking the antecedent (when present) is essential, as it provides valuable information for analyzing the phenomenon. In line with this consideration, we propose annotating the antecedent explicitly. More specifically, we suggest using the Misc column[23] to indicate whether the **antecedent** is present, or not (`Antec=Yes`; `Antec=No`) and its position in the text, identified by a **unique_number** (`AntecPosit=[unique_number]`). The unique_number has been introduced to identify nodes independently of their position in the text, since, as shown in Section 2, the

[20] For an alternative proposal for annotating ellipsis in basic UD, see the abstract at the following site: `https://unidive.lisn.upsaclay.fr/lib/exe/fetch.php?media=meetings:general_meetings:3rd_unidive_general_meeting:23_how_to_ellipsis_a_proposal_.pdf`

[21] Note that the `orphan` relation does not account for all cases of predicate ellipsis. It only highlights instances where a predicate is omitted and at least two remnants are present. Instances where the predicate is omitted but only one remnant survives are not marked with `orphan`; instead, they are annotated through promotion and thus may be overlooked.

[22] For the notion of parallelism in ellipsis: Phillips and Parker (2014, p.79)

[23] The Misc column is the last column of the CoNLL-U format used for annotation: `https://universaldependencies.org/format.html`.

antecedent can also be in a different sentence. It is displayed as a numeric value in the Misc column, and its ordering is not limited to the sentence but extends across the entire text/treebank. In A, we provide examples of its usage for *Italian-Old*. In cases of node splitting or duplication, the split node will receive a decimal number to avoid modifying annotation that have already been completed. For completeness of information and to reflect the reasoning adopted by the annotator, in cases where the antecedent is absent in the text, the AntecPosit value is annotated as "wk", indicating "world knowledge") (`AntecPosit=wk`). For instance, the Misc field of the empty node in the sentence that involves ellipsis will be the following: `Antec=Yes|AntecPosit=2`. This means that the antecedent is present (`Antec=Yes`), and that its position has the unique_token number 2 (`AntecPosit=2`).

- **what, if anything, should we reconstruct?** In an effort to balance the principle that more information enhances analysis, with the understanding that this is a preliminary step pending further refinement—and in the spirit of gathering community feedback and cross-linguistic analysis cases[24]—we suggest, at this initial stage, reconstructing the elided material, including—when permitted by the context (refer to Example 6 Subsection 7.2 for a specific issue)—its form, lemma, UPOS, features, head, and dependency relation. However, at the current stage of this study, we have decided not to address the reconstruction of arguments of the elided predicate,[25] even in the case of complex predicates, such as in an expression like "to make shield of". In such cases, only the predicate "make" is reconstructed. The decision to limit the reconstruction to verbs—while excluding arguments and complex predicates—is motivated, among others, by the difficulty of deriving valency information from UD annotation, given that the distinction between arguments and adjuncts is not explicitly encoded (De Marneffe et al., 2021, p. 13). We will provide examples in Section 7.

7 A Case-Study: Predicate Ellipsis in Italian-Old Treebank. Some Preliminary results and considerations

In this Section, we present examples of the enhancement of **predicate ellipsis** in a portion of the Italian-Old treebank, following the method described in Section 6.

Italian-Old[26] is a native UD treebank containing the *Divine Comedy* by Dante Alighieri, an Old Italian poem written between approximately 1306 and 1321. The poem is divided into three *Cantiche*: *Inferno* (Hell), *Purgatorio* (Purgatory), and *Paradiso* (Heaven). The first *Cantica*, *Inferno*, was manually annotated from scratch with respect to syntax. In contrast, the other two *Cantiche* were pre-parsed using a model trained on *Inferno* data and subsequently manually corrected (Corbetta et al., 2023).

The enhancement discussed here focuses on the first *Cantica*, *Inferno*, which comprises 33,416 tokens (excluding punctuation marks). In the following Subsections, we first describe the extraction of predicate ellipsis (Subsection 7.1), and then (Subsection 7.2) report on some noteworthy cases of enhancement in light of the proposals presented in Subsection 6.2.

7.1 Extraction

As a treebank natively annotated in UD, *Inferno* encodes ellipsis according to the UD guidelines, using the `orphan` relation and the promotion mechanism (see Section 3.1). To extract instances of predicate ellipsis, we queried for occurrences of the `orphan` relation and for all instances in which a non-verbal node (based on PoS) serves as the head of a dependency relation typically associated with verbal predicates, namely, nodes functioning as clause heads. We selected the following dependency relations: `root`, `parataxis`, `advcl`, `acl`, `ccomp`, and `csubj`, including their subtypes, if present.[27]

[24] As highlighted in Section 8, this work also aims at encouraging participation on this topic across different languages and to gather supporting evidence accordingly, with the goal of enriching the proposal and making it as language-independent as possible. To this end, we have undertaken preliminary experimentation with Latin treebanks.

[25] This choice contrasts with the approach adopted in the PDT style, which also reconstructs arguments `https://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/t-layer/html/ch06s12s01.html#elipsa1.1`).

[26] `https://github.com/UniversalDependencies/UD_Italian-Old`.

[27] We excluded `advcl:pred` and `xcomp` from the selection, as they are used for secondary predication, which we did not consider as instances of ellipsis. Moreover, we will not address `conj` in the current discussion, as it represents a broad topic that cannot be fully addressed within the scope of this paper.

Each case was then manually inspected to determine whether the retrieved instances genuinely represented ellipsis. With only a few exceptions, identified as annotation errors, all the examples retrieved by the query can be interpreted as valid instances of ellipsis.

In A, we report Table 1, representing the queries, and Table 2,[28] which reports the number of occurrences distributed across the selected deprels.

While a thorough analysis of the various types of ellipsis identified across the cases falls outside the scope of this study, it remains a goal for future research, as mentioned in 8.

7.2 Specific cases

When addressing the reconstruction of certain instances of ellipsis, we encounter several of the issues outlined in Subsection 6.2. In what follows, we illustrate some problematic cases related to the reconstruction process and the position of the reconstructed material (7.2.1) and the retrieval of the antecedent (7.2.2), and we describe the strategies we adopt to address them.

7.2.1 Reconstruction and position

Regarding the position of the ellipsis site and the material to be reconstructed, we present the following Example 4:

> **Example 4** - *Inf.* II, vv. 88-90
> *Temer si dee di sole quelle cose/ c'hanno potenza di fare altrui male;/ de l'altre no, ché non son paurose.*
> "Of those things only should one be afraid/ Which have the power of doing others harm;/ Of the rest, no; because they are not fearful."

In this example, the elliptical sentence is *de l'altre no* ("of the rest, no") and the antecedent is *Temer si dee* ("one should be afraid").

Even though in Example 4 the antecedent includes the lexical infinitive verb (*temer* "be afraid"), its modal verb (*dee* "should") and the reflexive clitic (*si* "one"), we reconstruct only a single node—specifically, the one corresponding to the content word—in line with the prioritization of content words over function words in UD.[29] This choice, however, does not preclude indicating the full antecedent information, which is conveyed by the unique_token attribute.[30] In A, we report the syntactic tree with enhanced dependency.

7.2.2 Antecedent retrieval

Another complex example involving the identification of the antecedent is reported in Section 3.1 (repeated here for clarity):

> **Example 5** - *Inf.* IV, vv. 31-32
> *Lo buon maestro a me: «Tu non dimandi/ che spiriti son questi che tu vedi?*
> "To me the Master good: "Thou dost not ask/ What spirits these, which thou beholdest, are?"

In this example, the ellipsis concerns the main clause *Lo buon maestro a me:* and consists in the omission of a *verbum dicendi*, that is, a verb whose meaning conveys an act of speaking. In this case, no explicit antecedent is retrievable either from the sentence itself or from the preceding context, as the previous dialogic exchange is also introduced through an elliptical construction (Example 6):

> **Example 6** - *Inf.* IV, vv. 19-21
> *Ed elli a me: «L'angoscia de le genti/ che son qua giù, nel viso mi dipigne/ quella pietà che tu per tema senti.*
> "And he to me: "The anguish of the people/ Who are below here in my face depicts/ That pity which for terror thou hast taken."

Example 6, however, does have a clear antecedent in vv. 16–17: *E io, che del color mi fui accorto,/ dissi:* "And I, who of his colour was aware,/ Said:".

Unlike the ellipsis in Example 6, where the antecedent can be retrieved just a few verses earlier (in v. 17), Example 5 lacks an antecedent in the immediate context and must therefore be interpreted through world knowledge. We report the annotation of Example 5 and 6 in A. In Example 6, both the form and lemma are reconstructed, as the

[28] The asterisk (*) following a deprel indicates that its subtypes were also included in the count. More specifically, `advcl` was queried along with `advcl:cmp`, `csubj` with `csubj:pass`, `acl` with `acl:relcl` and `ccomp` with `ccomp:reported`.

[29] `https://universaldependencies.org/u/overview/syntax.html#the-primacy-of-content-words`

[30] We note that the elided sentence also contains an instance of nominal ellipsis, namely *de l'altre [cose] no* (lit. "of the other [things] not"). However, since the present study focuses on predicate ellipsis, we do not reconstruct it.

antecedent is explicitly present. In contrast, in Example 5, the form is not specified; only the lemma is provided, as it is inferred.

8 Conclusion and Future Work

In this paper, we focus on a specific syntactic phenomenon, ellipsis, that has been analyzed across various frameworks, though to a lesser extent within the dependency-based tradition.

The inherent difficulty of capturing the nature of an omission in the text is compounded by the challenges of representing it graphically in syntactic corpora, namely treebanks. Within the state-of-the-art dependency framework (UD), ellipsis appears to be only marginally addressed, largely due to the annotation choices made at the basic level and the complexity involved in automatically retrieving such structures.

To address the lack of gold-annotated data, which are crucial for both linguistic analysis and machine learning, in this work we experimented on predicate ellipsis, as a preliminary step towards the ultimate goal of developing language-independent guidelines for the treatment of ellipsis in Universal Dependencies. Given the complexity of the topic, in this paper we have narrowed the scope to predicate ellipsis, with the aim of extending the analysis and annotation to all types of ellipsis in future works. For this paper, we focus on two complementary workflows: a theoretical one, centered on the analysis of ellipsis, and a practical one, aimed at providing a comprehensive annotation of predicate ellipsis in EUD. Examples of reconstructions, along with practical annotation cases, are presented in the final Section with respect to the UD treebank Italian-Old. An enhanced annotation of predicate ellipsis will be provided in the next release of the treebank Italian-Old. Given the inherently non-lexicalized nature of ellipsis, the proposed workflow can be extended to other languages through the use of morpho-syntactic annotation.
Concerning future work, a data-driven classification of predicate ellipsis in Italian-Old treebank is envisaged, one that emerges directly from the data and provides examples of ellipsis in context.

Additionally, we plan to develop a rule-based script, designed to be as language-independent as possible, to support the semi-automatic enhancement of predicate ellipsis. This step does not aim at a fully automatic resolution of ellipsis, which is still far from happening, but rather at facilitating manual annotation, which remains necessary. We welcome and encourage contributions from other treebank maintainers—or from researchers interested in exploring ellipsis—who wish to enrich their treebanks with this type of information.

Acknowledgments

This work received support from the CA21167 COST action UniDive, funded by COST (European Cooperation in Science and Technology). We would like to thank Daniel Zeman, Petya Osenova, and Simov Kiril for their valuable feedback throughout this research. Any remaining errors are solely the responsibility of the authors.

References

Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*. John Wiley & Sons.

Damir Ćavar, Ludovic Mompelat, and Muhammad Abdo. 2024a. The typology of ellipsis: a corpus for linguistic analysis and machine learning applications. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 46–54.

Damir Ćavar, Zoran Tiganj, Ludovic Veta Mompelat, and Billy Dickson. 2024b. Computing ellipsis constructions: Comparing classical nlp and llm approaches. In *Proceedings of the Society for Computation in Linguistics 2024*, pages 217–226.

Claudia Corbetta, Marco Passarotti, Flavio Massimiliano Cecchini, and Giovanni Moretti. 2023. Highway to hell. towards a universal dependencies treebank for dante alighieri's comedy.

Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational linguistics*, 47(2):255–308.

Kira Droganova, Filip Ginter, Jenna Kanerva, and Daniel Zeman. 2018. Mind the gap: Data enrichment in dependency parsing of elliptical constructions. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 47–54.

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*.

Jan Hajič, Eva Hajičová, Marie Mikulová, and Jiří Mírovskỳ. 2017. Handbook on linguistic annotation, chapter prague dependency treebank.

Jorge Hankamer and Ivan Sag. 1976. Deep and surface anaphora. *Linguistic inquiry*, 7(3):391–428.

Philip Hofmeister. 2007. Memory retrieval effects on filler-gap procession. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29.

Richard Hudson. 2010. *An introduction to word grammar*. Cambridge University Press.

Christopher Kennedy. 2003. Ellipsis and syntactic representation. In *The interfaces: Deriving and interpreting omitted structures*, pages 29–53. John Benjamins Publishing Company.

Thomas S. Kuhn. 1962. *The Structure of Scientific Revolutions*. University of Chicago Press.

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

Marjorie J McShane. 2005. *A theory of ellipsis*. Oxford University Press.

Jason Merchant. 1999. *The syntax of silence: Sluicing, islands, and identity in ellipsis*. University of California, Santa Cruz.

Jason Merchant. 2001. *The syntax of silence: Sluicing, islands, and the theory of ellipsis*. Oxford University Press.

Jason Merchant. 2018. Ellipsis: A survey of analytical approaches. In Jeroen van Craenenbroeck and Tanja Temmerman, editors, *The Oxford Handbook of Ellipsis*, Oxford Handbooks. Oxford University Press.

Marie Mikulová. 2014. Semantic representation of ellipsis in the prague dependency treebanks. In *Proceedings of the 26th Conference on Computational Linguistics and Speech Processing (ROCLING 2014)*, pages 125–138.

Iván Ortega-Santos, Masaya Yoshida, and Chizuru Nakao. 2014. On ellipsis structures involving a wh-remnant and a non-wh-remnant simultaneously. *Lingua*, 138:55–85.

Timothy Osborne. 2019. Ellipsis. In *A Dependency Grammar of English*, pages 349–378. John Benjamins Publishing Company.

Petya Osenova and Kiril Simov. 2003. The bulgarian hpsg treebank: Specialization of the annotation scheme. In *Proceedings of The Second Workshop on Treebanks and Linguistic Theories; Växjö, Sweden.*

Colin Phillips and Dan Parker. 2014. The psycholinguistics of ellipsis. *Lingua*, 151:78–95.

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

Martin Popel, Zdeněk Žabokrtskỳ, and Martin Vojtek. 2017. Udapi: Universal api for universal dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 96–101.

John Robert Ross. 1969. Guess who? In *Proceedings from the annual meeting of the chicago linguistic society*, volume 5, pages 252–286. Chicago Linguistic Society.

Maria Simi, Simonetta Montemagni, et al. 2018. Bootstrapping enhanced universal dependencies for italian. In *CEUR WORKSHOP PROCEEDINGS*, volume 2253. CEUR-WS.

Jeroen Van Craenenbroeck and Tanja Temmerman. 2018. *The Oxford handbook of ellipsis*. Oxford University Press.

A Appendix

1) **Orphan detection query**: pattern { N1 -[orphan]->N2 }

2) **Promotion detection queries**:
pattern {N1 -[deprel*]-> N2}
without {N2 [upos="VERB"] }
without {N2-[cop]->X}
without {N2-[orphan]->X}
without { N2 [upos="AUX"] }
 	
deprel* = root, parataxis, advcl, advcl:cmp, acl, acl:relcl, ccomp, ccomp:reported, csubj, and csubj:pass.

Table 1: Queries for Predicate Ellipsis

deprel	**occurrences**
orphan	124
root	36
parataxis	12
advcl*	95
ccomp*	19
acl*	0
csubj*	0

Table 2: Occurrences of Predicate Ellipsis in *Inferno*

Figure 1: Enhanced tree of Example 3

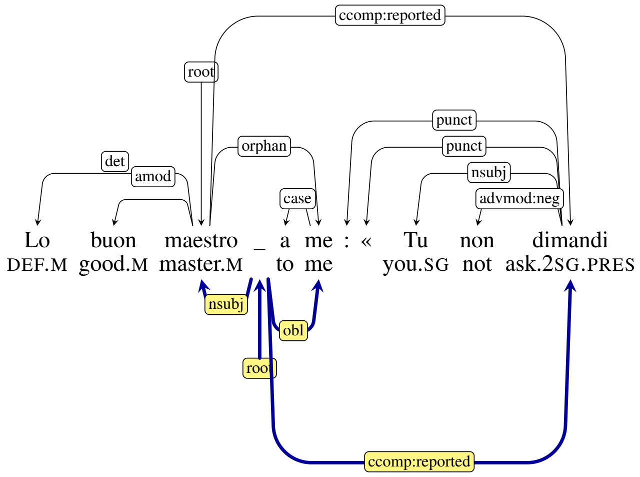

ccomp:reported
root
punct
punct
orphan
nsubj
det
amod
case
advmod:neg
Lo buon maestro _ a me : « Tu non dimandi
DEF.M good.M master.M to me you.SG not ask.2SG.PRES
nsubj
obl
root
ccomp:reported

Figure 2: Enhanced tree of Example 4: antecedent in the same sentence. *For reasons of space, we do not report the unique_number (UN) for each word.

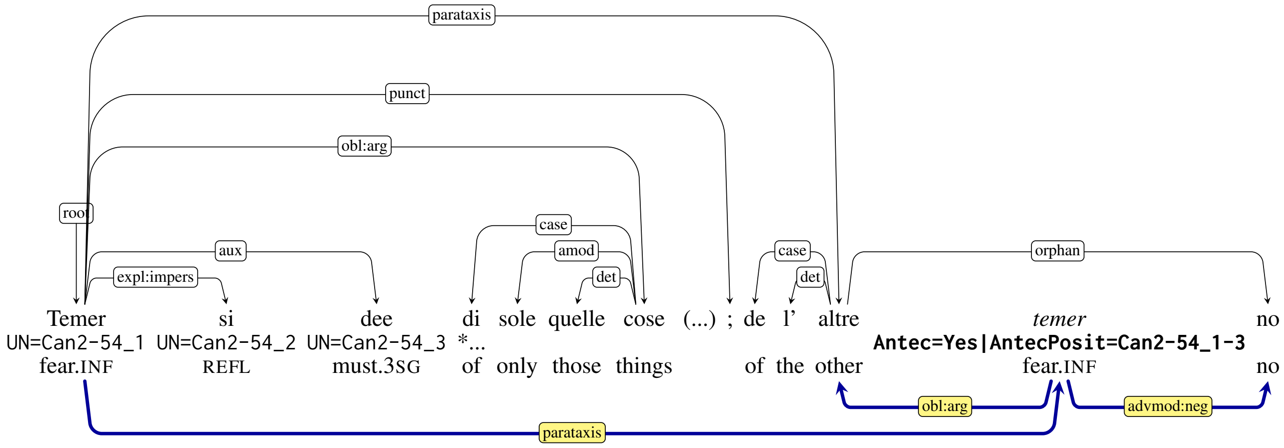

parataxis
punct
obl:arg
root
case
aux
amod
case
orphan
expl:impers
det
det
Temer si dee di sole quelle cose (...) ; de l' altre temer no
UN=Can2-54_1 UN=Can2-54_2 UN=Can2-54_3 *...
Antec=Yes|AntecPosit=Can2-54_1-3
fear.INF REFL must.3SG of only those things of the other fear.INF no
obl:arg
advmod:neg
parataxis

Figure 3: Enhanced tree of Example 5: antecedent not overtly present in the text.

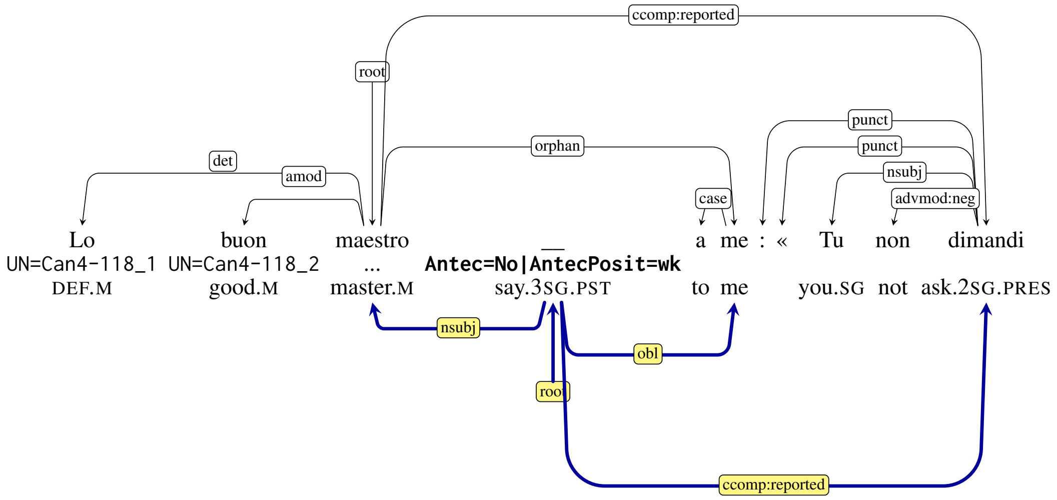
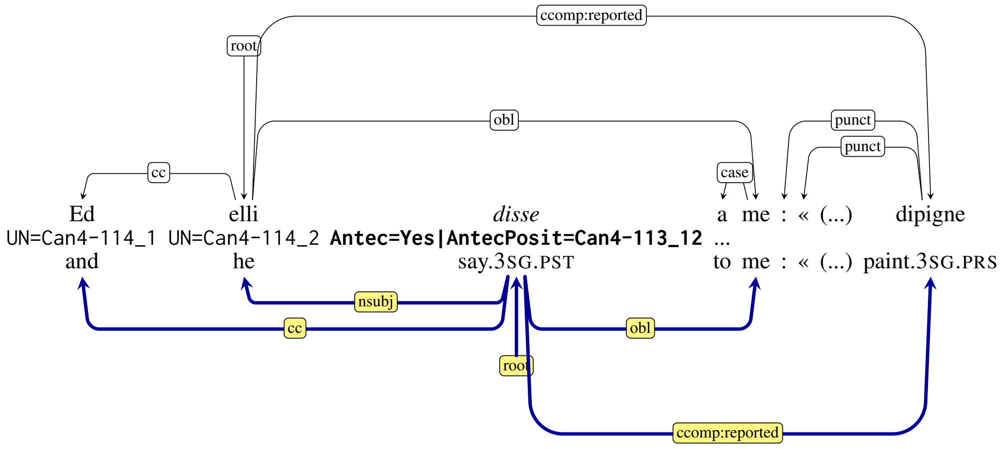

ccomp:reported
root
punct
orphan
punct
det
amod
nsubj
case
advmod:neg
Lo buon maestro _ a me : « Tu non dimandi
UN=Can4-118_1 UN=Can4-118_2 ... Antec=No|AntecPosit=wk
DEF.M good.M master.M say.3SG.PST to me you.SG not ask.2SG.PRES
nsubj
obl
root
ccomp:reported

Figure 4: Enhanced tree of Example 6: antecedent not in the same sentence.


ccomp:reported
root
obl
punct
punct
cc
case
Ed elli disse a me : « (...) dipigne
UN=Can4-114_1 UN=Can4-114_2 Antec=Yes|AntecPosit=Can4-113_12 ...
and he say.3SG.PST to me : « (...) paint.3SG.PRS
nsubj
cc
obl
root
ccomp:reported

Case Syncretism in Kasavakan Puyuma: A Field Data Analysis of Noun Phrase Markers

Deborah Watty[1], Yung-Jui Yao[1], Jens N. Watty[2]

[1]Graduate Institute of Linguistics, National Taiwan University
[2]Department of Physics, National Taiwan University
{r11142012, r10142002, r11222082}@ntu.edu.tw

Abstract

Previous research has reported differing patterns of case syncretism across three dialects of Puyuma, an Austronesian language of Taiwan (Nanwang, Katipul, Ulivelivek). This study presents a quantitative analysis of case syncretism of noun phrase markers and disambiguation strategies in the Kasavakan dialect. Our dataset comprises 377 sentences elicited from five speakers, which we annotated for voice, potential semantic ambiguity, word order, and case marking of different semantic roles. We find evidence for a high degree of syncretism between genitive and nominative markers, alongside a decline in the use of genitive forms, particularly for common definite nouns. Some overlap with oblique markers is also attested, suggesting varying degrees of case syncretism between speakers. Topicalization appears to be the most frequent disambiguation strategy, while the order of non-topicalized noun phrases does not seem to aid disambiguation. Other factors, including age and individual experiences may contribute to inter-participant variation. These findings contribute to a more complete understanding of case marking in Puyuma by adding new empirical data from the Kasavakan dialect, where patterns of syncretism and disambiguation differ from previously described varieties.

1 Introduction

Puyuma is a Formosan language spoken primarily in Taitung County in southeastern Taiwan by the Puyuma people, whose population is approximately 13,000 (Teng, 2018). It is particularly relevant to the reconstruction of Proto-Austronesian, as it has been argued to represent one of the primary branches of the Austronesian language family (Ross, 2009). Traditionally, the Puyuma community consists of eight main villages: Puyuma (Nanwang), Katipul, Rikavung, Tamalakaw, Kasavakan, Pinaski, Alipai, and Ulivelivek (Teng, 2009, 2018). For educational purposes, the language is classified into four dialects: Puyuma (Nanwang), Katipul, Kasavakan, and Ulivelivek (Teng, 2018).

Similar to other Austronesian languages, central features of Puyuma syntax are its use of noun phrase markers (NPMs) and the distinction between actor voice and undergoer voice. In actor voice (AV), the subject of the verb is the actor (Example 1). In undergoer voice, an argument other than the actor becomes the subject, while the actor is relegated to non-subject status. The three variants of undergoer voice are Patient Voice (PV, Example 2), Locative Voice (LV, Example 3), and Conveyance Voice (CV, Example 4):

1. (Nanwang, Teng, 2008)[1]

tr<em>akaw	dra	paisu	i	Isaw
<AV>steal	OBL.INDF	money	NOM.SG	Isaw

Isaw stole money.

2. (Nanwang, Teng, 2008)

tu=trakaw-aw	na	paisu	kan	Isaw
3SG.GEN=steal-PV	NOM.DEF	money	OBL.SG	NAME

Isaw stole the money.

3. (Nanwang, Teng, 2008)

tu=trakaw-ay=ku	dra	paisu	kan	Isaw
3SG.GEN=steal-LV=1SG.NOM	OBL.INDF	money	OBL.SG	NAME

Isaw stole money from me.

4. (Nanwang, Teng, 2008)

tu=trakaw-anay	i	tinataw	dra	paisu
3SG.GEN=steal-CV	NOM.SG	his.mother	OBL.INDF	money

He stole money for his mother.

[1] In examples from other papers, we have changed the glossing conventions to our own for the sake of comparability. A list of abbreviations is included in Appendix A.

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 64–73
August 28-29, 2025 ©2025 Association for Computational Linguistics

Note how in the above examples, all of which are from the Nanwang dialect, the non-subject actor is marked with an oblique NPM (see Examples 2 and 3). This is not always the case in the Katipul and Ulivelivek dialects, which sometimes use a genitive marker (Teng, 2009). In the Nanwang dialect, genitive and oblique NPMs are fully syncretic (see Figure 1).

While the use of NPMs has been documented for the other three dialects, the Kasavakan dialect remains understudied in this regard. To provide a more comprehensive understanding of Puyuma NPMs, it is of research interest to document the Kasavakan NPMs.

We initially collected an explorative data set from only one speaker who seemed to display a distinct pattern of case syncretism, as shown in Example 5:

5. (Kasavakan, Speaker 1)
   tu=pa-ated-ay na pawko
   3SG.GEN=CAUS-send-LV NOM.DEF package
   i Dipung *i* Simuy
   LOC Japan NOM.SG NAME
   Simuy sent a package to Japan.

Here, the non-subject actor, Simuy, is marked with a nominative NPM, suggesting that case syncretism might go in the opposite direction of Nanwang, with nominative and genitive being syncretic. However, data collected from four additional speakers showed a complex and variable pattern with notable inter-speaker variability. A quantitative approach was thus adopted to account for the nature of the distribution of these markers more effectively.

The goals of this study were as follows:

1. Describe the use of NPMs in Kasavakan Puyuma, an aspect not previously studied.
2. Identify preferred disambiguation strategies in cases of syncretism.

This paper is structured as follows. Section 2 reviews prior work by Teng (2009), which describes NPM use and disambiguation strategies in Nanwang, Ulivelivek, and Katipul Puyuma. Section 3 outlines the data collection and annotation process. Section 4 presents our findings, including the analytical approach and illustrative examples from our dataset. Section 5 discusses the implications of the results and offers additional observations.

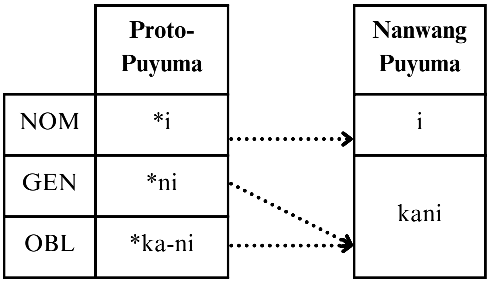


Figure 1: Illustrative example of genitive-oblique case syncretism in Nanwang Puyuma (personal singular genitive marker), based on Teng (2009).

2 Literature Review

To our knowledge, Teng (2009) is the only work that covers the differences between the patterns of case syncretism between different dialects of Puyuma in detail. Citing Baerman et al. (2005), Teng distinguishes between diachronic and synchronic syncretism, the former referring to forms being merged over time so that the distinction between the two forms disappears, and the latter referring to one form covering two functions in certain cases whereas those functions have separate forms elsewhere in the language.

Teng's reconstruction of Proto-Puyuma NPMs contains two genitive markers: *ni* for personal singular nouns and *nina* for common definite nouns. In the case of Nanwang Puyuma, Teng argues that the genitive case has completely syncretized with the oblique case, resulting in a pattern where rather than a three-way distinction between subjects (nominative), non-subject actors and possessors (genitive) and other non-core arguments (oblique), the distinction is now between subjects (nominative) and non-subjects (oblique). As illustrated in Figure 1, this has resulted in genitive markers becoming obsolete.

In Katipul, which is geographically closest to Kasavakan, genitive markers are replaced by nominative markers rather than oblique markers for common definite nouns, such as in Example 6:

6. (Katipul, Teng, 2009)
   tu=atek-aw na sa'az
   3SG.GEN=hack-PV NOM.DEF branch
   na lakak
   NOM.DEF children
   The children hacked the branches.

Rather than a distinction between subject and non-subject as in Nanwang, the new distinction in this case appears to be whether or not the noun phrase is a core argument, with nominative NPMs being used to mark all core arguments including subjects, possessors, and non-subject actors.

However, nominative-genitive syncretism is not complete in Katipul. Genitive markers have not been lost entirely and tend to be used for disambiguation, as in Example 7:

7. (Katipul, Teng, 2009)
tu=karatr-aw na suan *nina*
3SG.GEN=bite-PV NOM.DEF dog GEN.DEF
unan
snake
The snake bit the dog.

In Example 6, the semantics of the verb alone are sufficient to disambiguate the actor, whereas in Example 7, using nominative *na* to mark both nouns would result in the sentence being semantically ambiguous. Using the genitive marker *nina* clearly marks the snake as the actor in this sentence.

The fact that genitive markers have not been entirely replaced by nominative markers in Katipul Puyuma becomes even more apparent when considering personal singular nouns, where a distinction between nominative and genitive is obligatory. Interestingly, common indefinite nouns show the same genitive-oblique syncretism as seen in Nanwang Puyuma.

The situation is similar in Ulivelivek Puyuma, with one difference being that genitive-oblique syncretism also applies to common definite non-subject actors (see Example 8), but not to possessors (see Example 9):

8. (Ulivelivek, Teng, 2009)
tu=senan-ay
3SG.GEN=sunburned-LV
*nina/kana/*na* kadaw
GEN/OBL/*NOM.DEF sun
It was burned by the sun.

9. (Ulivelivek, Teng, 2009)
tu=tial *nina/na/*kana* suan
3SG.GEN=belly GEN/NOM/*OBL.DEF dog
the dog's belly

Teng's findings can be summarized as follows:

- In Nanwang Puyuma, genitive and oblique markers are fully syncretic, i.e., oblique markers have completely replaced possessive and genitive markers, leading to ambiguities between non-subject actors and other oblique noun phrases.
- In the Katipul dialect, nominative and genitive/possessor markers are partially syncretic for common definite nouns, leading to ambiguities between actor and subject in undergoer voice. This type of ambiguity can be resolved by using the more specific genitive marker or through topicalization. The distinction between nominative and genitive is obligatory for personal nouns. For common indefinite nouns, the pattern is the same as in the Nanwang dialect.
- Only the Ulivelivek variety distinguishes between markers for non-subject actors and possessors in some cases. While a specific genitive marker exists as in the Katipul dialect, genitive NPMs are partially syncretic with oblique NPMs, and possessor NPMs are partially syncretic with nominative NPMs.
- Preferred disambiguation strategies vary by dialect as ambiguities arise in different situations. Depending on the dialect, strategies can include topicalization, word order, verbal semantics and cross-referencing.

3 Dataset

The dataset[2] we present was selected from sentences collected during interviews with five different speakers of Kasavakan Puyuma (4 female and 1 male, born between 1953 and 1958). All interviews took place in the first half of 2024 as part of a class on field linguistics. Communication with the speakers was conducted through Mandarin. Speakers were interviewed separately so that they could not directly comment on each other's sentences. The final dataset contains 377 sentences, some of which were directly elicited, while others were presented to the speaker and rated acceptable or unacceptable (see Figure 2).

The criteria for inclusion of a sentence in the final dataset were as follows:

- The sentence contains a semantically transitive verb and/or a possessive structure.
- No pronouns except for the genitive clitic *tu* in undergoer voice sentences.

[2] https://github.com/deborahwatty/kasavakan_npms.git

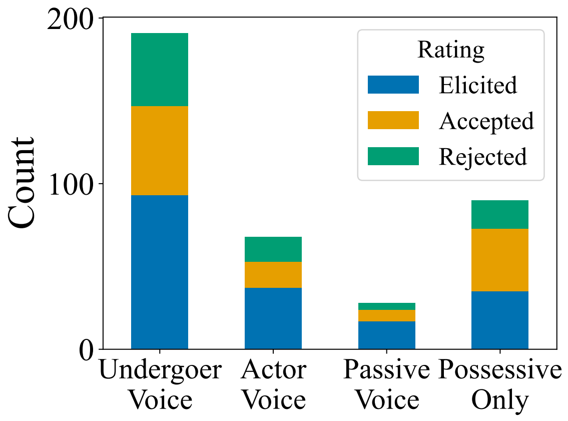


Figure 2: Overview of the number of sentences by voice and rating. The dataset also includes entries that only consist of a possessive construction.

- Sentence structure not too complex, no existential clauses, no copula.
- We excluded sentences where the speaker was unsure about the meaning or correctness.

Annotation was performed by consensus among the authors. Each sentence in the dataset was annotated for the following features:

Rating

We annotated whether sentences were *directly elicited*, suggested by us and rated *acceptable*, or *rejected* by the speakers.

Voice

This feature denotes whether the sentence is in *actor voice*, *passive* or one of the undergoer voices (*patient*, *locative*, or *conveyance*). When one of the semantic roles is filled by a possessed noun, this is also annotated in this column.

Semantic Roles

The possible semantic roles in the dataset are *actor*, *undergoer*, *beneficiary*, *location*, *possessor* and *theme*. For each sentence, we annotated the type (*personal* or *common* noun, *definite* or *indefinite* for common nouns, and *singular* or *plural* for personal nouns) as well as the grammatical case of the noun that fills each semantic role. Note that for common definite nouns, the annotation does not distinguish between cases where an NPM was used and cases where a demonstrative was used (such as *ini* or *kanidu*, see Example 21). These cases are counted as variations of the nominative marker *na* or oblique marker *kana*, respectively, which is the approach taken by Teng (2009).

Ambiguity

This feature describes which of the semantic roles in a given sentence may be confused for each other when given only the verb in its basic form and the words that fill the different semantic roles, but not the word order or further context. For example, given the words “to eat”, “cookie” and “Tom”, it would be reasonable to assume that “Tom” is the actor and the “cookie” is the undergoer with little room for ambiguity. However, when given the words “to bite”, “snake” and “dog”, it is unclear which of the animals is the actor and which is the undergoer. In this case, we would annotate *actor-undergoer ambiguity*.

Word Order

We also annotated the order in which the different constituents of the sentence are arranged. For example, the word order of Example 12 is annotated as *Verb-Undergoer-Beneficiary-Actor*. *Possessors* and *possessed nouns* are annotated individually; where these structures are described by a *predicate* (such as “is thick” or “are round”), the predicate is denoted as such. Where there are additional constituents that do not fit into any of the aforementioned semantic roles, the letter *X* denotes the presence of additional constituents at this position in the sentence.

Topicalization

Here, we annotated which, if any, of the constituents are topicalized (i.e., placed at the beginning of the sentence, before the verb).

4 Analysis and Results

4.1 Distribution of Noun Phrase Markers

The distribution of NPMs in our dataset is compared to the data by Teng (2009, 2018) in Table 1.

Below follows a more detailed description of the data used to fill in the fields for possessors (PSR) and non-subject actors (GEN) in the Kasavakan column. As all topicalized nouns were consistently marked as nominative (with a single exception, all sentences where this was not the case were rejected by all speakers), examples where the noun in question is topicalized are excluded from this part of the analysis. Post-hoc two-sided binomial tests (Siegel and Castellan, 1988) were conducted to examine whether speakers showed a tendency to prefer any specific marker.

			Proto-Puyuma	Nanwang	Katipul	Ulivelivek	Kasavakan
Personal	Singular	NOM	*i	i	i	i	i
		PSR	*ni	kan	ni	ni	i / ni / kani
		GEN					i / ni / (kani)
		OBL	*ka-ni		kani	kani	kani / i?
	Plural	NOM	*na	na	na	na	na
		PSR	(unknown)	kana	nina / na	(unknown)	na / kana
		GEN				(unknown)	na / (kana?)
		OBL	*ka-na		kana	kana	kana
Common	Definite	NOM	*na	na	na	na	na
		PSR	*ni-na	kana	na / nina	nina / na	na / kana
		GEN				nina / kana	na / kana
		OBL	*ka-na		kana	kana	kana
	Indefinite	NOM	*a	a	a	a	a
		PSR	*dra	dra	za	za	(da) / (a)
		GEN					a / (da)
		OBL					da

Table 1: Comparison of the distribution of NPMs across dialects. The data on the Nanwang, Katipul and Ulivelivek variants as well as the tentative reconstruction of Proto-Puyuma is taken from Teng (2009), the personal plural NPMs for Katipul come from Teng (2018), and the Kasavakan column is based on our dataset. Parentheses indicate that instances of the form were rated acceptable by speakers but not actively used by any speaker. Where multiple markers were acceptable, they are sorted by the perceived preference of the speakers overall. NPMs that only occurred once are marked with a question mark. The case column denotes the function that the NPMs take on in a sentence, whereas the glossing in examples in the main text denotes the form of the NPM – not the function.

4.1.1 Personal Singular Nouns

There is some flexibility in how personal singular possessors are marked. While the nominative marker was used most often in elicited sentences, the difference is not statistically significant (two-sided binomial test, nominative vs. non-nominative, $N = 19$, $p_0 = 1/3$, $p = 0.089$; Figure 3). Instances of all three markers are found in the data, although there seem to be differences between individual speakers. For example, Speaker 5 pointed out that she differentiates between possessive noun phrases to be used in the context of a longer sentence (such as Example 10) and the same phrase being used in isolation to express that an item belongs to the possessor (such as Example 11). In the former, any of the three markers are acceptable to her, but she would not use a nominative marker in the latter. Speaker 2 disagrees and used the nominative version *i Lutan* when asked how she would express “This car is Lutan’s” (as in Example 11). In the following, directly elicited NPMs are bolded.

10. (Kasavakan, Speaker 5)
tu=paliding **ni**/i/kani Lutan
3SG.GEN=car **GEN**/NOM/OBL.**SG** NAME
Lutan’s car

11. (Kasavakan, Speaker 5)
tu=paliding **ni**/kani/*i Lutan
3SG.GEN=car **GEN**/OBL/*NOM.**SG** NAME
This car is Lutan’s.

Personal singular non-subject actors also show flexible case marking, although not to the same extent as possessors. We observe a clear preference for nominative markers (two-sided binomial test, nominative vs. non-nominative, elicited only, $N = 18$, $p_0 = 1/3$, $p < 0.001$; see Figure 4). The dataset only contains one elicited sentence with a genitive non-subject actor (Example 12), which was provided by Speaker 5, who also rated the version with the nominative marker as acceptable:

12. (Kasavakan, Speaker 5)
tu=veray-ay na liwu kani
3SG.GEN=give-LV NOM.DEF gift OBL.SG
Lutan **ni**/i Pusang
NAME **GEN**/NOM.**SG** NAME
Pusang gives the gift to Lutan.

4.1.2 Personal Plural Nouns

Personal plural NPMs indicate that the referent of the personal noun includes not only the individual

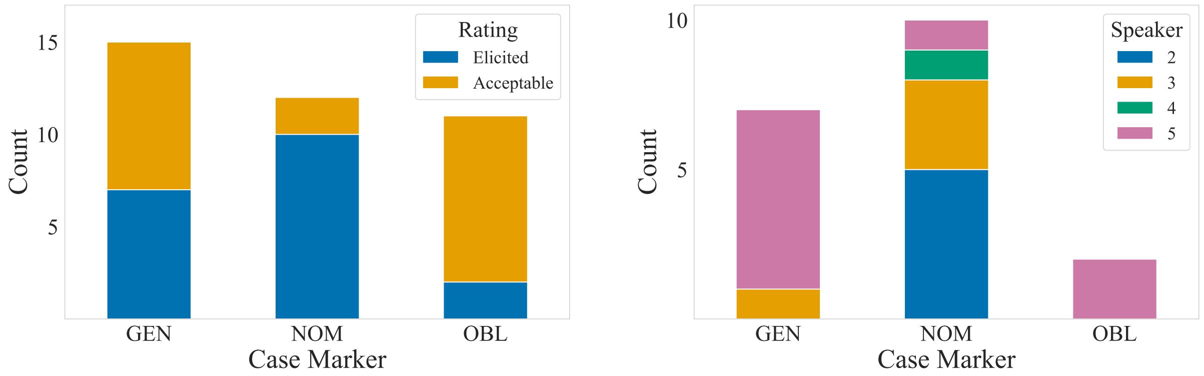


Figure 3: Distribution of NPMs for personal singular possessors, excluding topicalized noun phrases. **Left:** All acceptable sentences, divided by elicited vs. rated acceptable. **Right:** Only directly elicited sentences, divided by speakers.

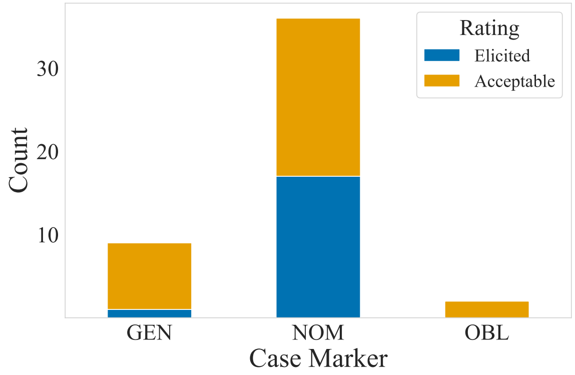


Figure 4: Distribution of NPMs for personal singular non-subject actors, excluding those in sentences where the actor is topicalized.

denoted by the noun, but also a broader group associated with that person. According to Speaker 3, the group may be their family, their friends or any other group of people they may be associated with.

All plural markers in the dataset were either nominative or oblique. Figure 5 shows the distribution of NPMs for personal plural possessors. Nominative markers are used most often but our data is too sparse to establish a clear preference (two-sided binomial test, nominative vs. oblique, elicited only, $N = 8$, $p_0 = 1/2$, $p = 0.070$). The lone actively used oblique marker (Example 13) was given as an alternative to the nominative marker:

13. (Kasavakan, Speaker 3)

nantu	paliding	**na/kana**
3SG.POSS.NOM	car	**NOM/OBL.PL**

Lutan
NAME
Lutan et al.'s car

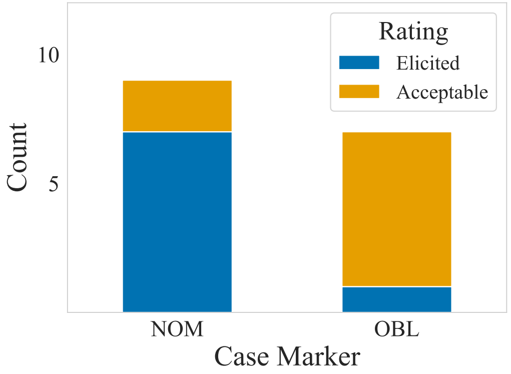


Figure 5: Distribution of NPMs for personal plural possessors, excluding topicalized noun phrases.

As for non-subject actors, the data is even sparser, but both instances of non-topicalized non-subject actors were marked with the nominative marker *na*, as in Example 14:

14. (Kasavakan, Speaker 5)

na	vulraw	tu=akan-aw	**na**
NOM.DEF	fish	3SG.GEN=eat-PV	**NOM.SG**

Umang
NAME
Umang et al. ate fish.

It is difficult to draw any conclusions about the acceptability of the oblique *kana* for non-subject actors as the dataset does not contain any non-topicalized occurrences. In topicalized sentences, we only have two contradicting examples of accepted (not elicited) sentences, Examples 15 and 16:

15. (Kasavakan, Speaker 3)
na/*kana Lutan (mu),
NOM/*OBL.**PL** NAME (TOP)
tu=veranay na kavang
3SG.GEN=gift.LV NOM.DEF clothes
kana lralrak
OBL.DEF children
Lutan et al. give the clothes to the children.

16. (Kasavakan, Speaker 2)
kana Umang mu tu=pa-akan-anay
OBL.PL NAME TOP 3SG.GEN=CAUS-eat-CV
idu na vulraw
that.NOM LNK fish
Umang et al. are feeding that/those fish.

We did not encounter a specific genitive marker in this category, and as Teng (2009) did not include data on a potential personal plural genitive marker in Katipul or Ulivelivek, there was no basis for us to ask about the acceptability of such a hypothetical marker.[3]

4.1.3 Common Definite Nouns

Speakers prefer to mark common definite possessors as nominative (two-sided binomial test, nominative vs. oblique, $N = 12$, $p_0 = 1/2$, $p = 0.006$), but often also accept oblique markers (same test, elicited and accepted combined, $N = 21$, $p = 0.189$; see Figure 6).

Similarly, for non-subject actors, speakers strongly prefer the nominative (same test, elicited only, $N = 17$, $p = 0.013$; see Figure 7). While oblique markers were actively used in three instances, it is worth noting that two of these examples were given by Speaker 5, whose preferences also seem to deviate from the other speakers in other cases (such as for personal singular nouns, see Figure 3).

The most striking finding is that the genitive marker *nina* was universally rejected, indicating that it is not used at all in Kasavakan.

4.1.4 Common Indefinite Nouns

NPMs for common indefinite nouns are some of the most difficult to elicit because Mandarin does not have a definite-indefinite distinction. Indefinite possessors were rated acceptable with both nominative and oblique markers (with two and three examples, respectively), but there were no instances of indefinite possessive markers in elicited sentences.

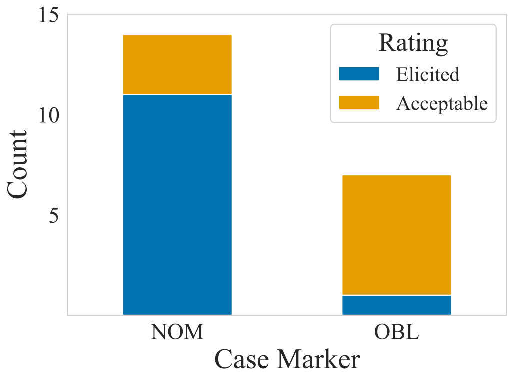


Figure 6: Distribution of NPMs for common definite possessors, excluding topicalized noun phrases.

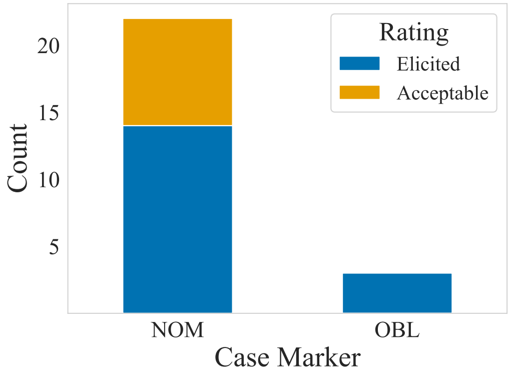


Figure 7: Distribution of NPMs for common definite non-subject actors, excluding those in sentences where the actor is topicalized.

For non-subject actors, there was only one case of an indefinite marker being actively used (Example 17), and it was one of several options given:

17. (Kasavakan, Speaker 5)
tu=aday-aw na kuce
3SG.GEN=take-PV NOM.DEF shoes
na/kana/a/da suwan
{**NOM/OBL**}.**DEF**/{**NOM**/OBL}.**INDF** dog
The shoes were moved by the/a dog.

The lack of elicited sentences containing indefinite NPMs makes it difficult to make definitive statements about the preferred marking of common indefinite nouns.

4.2 Disambiguation Strategies

4.2.1 Word Order

Topicalization is the only obvious and consistent disambiguation strategy in our data. An analysis

[3] The personal plural use of *nina*, as attested for Katipul by Teng (2018), was not known to us at the time of the interviews.

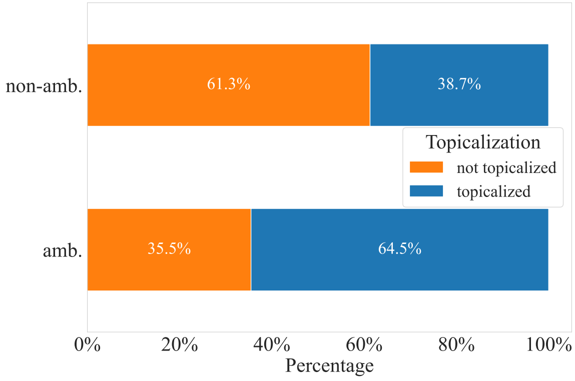


Figure 8: Topicalization by ambiguity status in elicited undergoer voice sentences (N=93). *amb.* refers to ambiguous and *non-amb.* refers to non-ambiguous.

of the order of non-topicalized noun phrases in elicited ambiguous undergoer voice sentences revealed no significant difference between the orders *verb-undergoer-agent* ($N = 4$) and *verb-agent-undergoer* ($N = 6$). Ambiguity can arise when two noun phrases marked with the same NPM appear on the same side of the verb (see Example 18).[4] Topicalizing the agent resolves the ambiguity (see Example 19).

18. (Kasavakan Speaker 5)

tu=karac-aw	**na**	unan
3SG.GEN=bite-PV	**NOM.DEF**	snake

kana	suwan
OBL.DEF	dog

The dog bit the snake./The snake bit the dog.

19. (Kasavakan Speaker 3)

na	unan	tu=karac-aw
NOM.DEF	snake	3SG.GEN=bite-PV

na	suwan
NOM.DEF	dog

The snake bit the dog.

The ratio of topicalized vs. non-topicalized sentences in ambiguous and non-ambiguous sentences is visualized in Figure 8. A Fisher's exact test (Fisher, 1922) revealed that ambiguous sentences were significantly more likely to be topicalized than unambiguous ones (odds ratio = 2.88, $p = 0.026$).

Further evidence that topicalization is used as a disambiguation strategy comes from the fact that in ambiguous and unambiguous sentences alike, it is almost universally the actor (and only rarely the undergoer) that is topicalized. Actor topicalization occurred in 36/40 ambiguous and 10/12 unambiguous topicalized undergoer voice sentences. In both cases, this preference was significant (two-sided binomial tests, $p_0 = 1/2$, $N = 40$, $p < 0.001$; $N = 12$, $p = 0.039$).

4.2.2 Avoiding Undergoer Voice

In some cases where we suggested ambiguous undergoer voice sentences, speakers actively gave alternative sentences in actor voice or passive, such as Examples 20 and 21:

20. (Kasavakan, Speaker 3)

t<em>enged	**i**	Pusang	**kani**
<AV>beat	**NOM.SG**	NAME	**OBL.SG**

Umang
NAME

Pusang beat Umang.

21. (Kasavakan, Speaker 3)

ini	na	suwan	ki-karac	**kana**
this.NOM	LNK	dog	PASS-bite	**OBL.DEF**

unan
snake

This dog was bitten by the snake.

In these versions, ambiguity is resolved because the actor is necessarily marked nominative in actor voice and necessarily marked oblique in passive constructions, making it unnecessary to use further disambiguation strategies.

5 Discussion

5.1 Implications

Our data reveals that non-subject actors and possessors tend to be marked nominative in Kasavakan Puyuma, but there are cases where genitive markers are still used, and even oblique markers are frequently found to be acceptable by speakers.

Nominative-genitive syncretism appears to be stronger in Kasavakan than in Ulivelivek and Katipul, as evidenced by the fact that genitive markers seem to not be used at all for common definite nouns. Additionally, since not all speakers use genitive markers actively, the use of genitive markers does not appear to be a preferred disambiguation strategy.

Speakers tend to disambiguate the semantic roles in ambiguous sentences through the use of topicalization. Alternatively, they may avoid using undergoer voice altogether and opt for actor voice or a passive construction instead.

[4] This sentence was elicited twice in different sessions for opposite elicitation prompts.

5.2 Potential Reasons for Variation

A potential explanation for the fact that oblique markers are sometimes accepted for non-subject actors may be that the speakers have had contact with other villages. We directly asked Speaker 5 about the use of *ni* (since she prefers to mark personal possessors with a genitive marker, see Figure 3). She mentioned that she was aware that *ni* is commonly used in Ulivelivek. Furthermore, she noted that the marker used to be more common in Kasavakan and is now generally used less frequently by those under the age of 80. If her observation is accurate, it may also be evidence for diachronic nominative-genitive syncretism in Kasavakan – a complete syncretism between nominative and genitive could be a possibility in the future.

Besides nominative-genitive syncretism, there are hints that the distribution of case markers may be even less constrained in Kasavakan. While there is no elicited sentence with the nominative marker *na* in the oblique sense, Speaker 4 accepted such a use in Example 22.[5] In addition, the personal singular nominative marker *i* was marked on the undergoer in a monotransitive causative sentence (see Example 23, compared to Example 24), a position where oblique is expected in the Katipul dialect (see Example 25). Additionally, we also observe possessive and genitive usage of *i* (see Examples 10 and 12, respectively) and *na* (see Examples 26 and 17, respectively).

22. (Kasavakan, Speaker 4)
    na suwan ’<em>a~’evang
    NOM.DEF dog <AV>RED~chase
    na tutus
    NOM.DEF mouse
    The dog is chasing the mouse.

23. (Kasavakan, Speaker 1)
    pa-uwa i Valraka t<em>akesi-a
    CAUS-go LOC USA <AV>study-PJ
    i Lutan
    NOM.SG NAME
    (He) let Lutan study in the US.

24. (Kasavakan, Speaker 1)
    pa-uwa i Valraka t<em>akesi-a ***kani***
    CAUS-go LOC USA <AV>study-PJ **OBL.SG**
    Lutan i malri
    NAME NOM.SG father
    The father let Lutan study in the US.

25. (Katipul Teng, 2018)
    pa-uwa=ku *kana* alrak i
    CAUS-go=1SG.NOM OBL.DEF child LOC
    palakuan
    palakuan
    I asked the child to go to the palakuan (adult assembly hall).

26. (Kasavakan Speaker 5)
    tu=sa’ad **na** kawi tatelraw
    3SG.GEN=branch **NOM.DEF** tree long
    The branches of the tree are long.

These phenomena point to the possibility that Kasavakan case markers have become more syncretic and that these markers are gradually losing their case-marking abilities. This would require greater use of additional disambiguation strategies, which may include topicalization, word order and verbal semantics. Further research is needed to confirm the oblique uses of these markers.

5.3 Conclusion

The first goal of this study was to describe the distribution of NPMs in Kasavakan Puyuma and identify patterns of case syncretism. The results, which are based on data collected from five speakers, are shown in Table 1.

Looking at the most preferred markers, nominative, genitive and possessive markers appear to be partially syncretic in the Kasavakan dialect. Common indefinite nouns may be an exception, but our data is inconclusive due to the limited number of such examples. It can be concluded that there is some flexibility in the use of NPMs and that individual preference plays a significant role. Overall, the distribution is closer to the Katipul and Ulivelivek varieties than to the Nanwang variety of Puyuma. While the genitive marker *nina* for common nouns seems to have been lost, some speakers still use the genitive marker *ni* for personal nouns.

The second objective was to identify the preferred disambiguation strategies for sentences where case syncretism causes ambiguities. Topicalization was shown to be a frequent strategy, with the use of actor voice or passive rather than undergoer voice being an alternative. The data was inconclusive on the role of the order of non-topicalized noun phrases in undergoer voice sentences.

In summary, this study provides an insight into the distribution of NPMs in Kasavakan Puyuma, the patterns of case syncretism, and the preferred disambiguation strategies.

[5] The undergoer in actor voice sentences is usually oblique.

Limitations

The results of this study are subject to a number of limitations that need to be addressed.

First, the current study uses sentences elicited from a limited number of speakers, all of whom belong to a very limited age group. This is due to the relatively small number of speakers and the lack of available data. The elicited sentences were designed to answer the research questions. Hence, the dataset may deviate from spontaneous speech.

Second, the English translations in our dataset typically correspond to the original Chinese prompts; however, in some cases where speakers retranslated a sentence they had produced, the translation was adjusted to reflect their retranslation. This cases are not explicitly marked as such in the dataset.

Third, as mentioned in the results section, the dataset has very few examples of some types of NPMs, making some of the entries in Table 1 less conclusive than others. This is in part due to the difficulty of eliciting some of the rarer NPMs.

Fourth, while we report uncorrected p-values here, we note that multiple comparisons were conducted, and results should be interpreted with appropriate caution. Additionally, as all annotations were reviewed collaboratively and ambiguous cases were resolved through discussion among the authors, we were unable to compute formal inter-annotator agreement, which would have yielded an objective estimate of annotation reliability.

Finally, some of our annotations may be open to debate, as we worked with a limited set of semantic roles that occasionally required applying role definitions somewhat broadly.

Acknowledgments

We would like to thank Professor Li-May Sung for her invaluable guidance throughout this project and for organizing the interviews with Puyuma speakers, without which this work would not have been possible. Any shortcomings of this paper are our own. We also appreciate the Puyuma informants who generously shared their knowledge of their language.

References

Matthew Baerman, Dunstan Brown, and Greville G Corbett. 2005. *The syntax-morphology interface: A study of syncretism*. 109. Cambridge University Press.

Ronald A. Fisher. 1922. On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the royal statistical society*, 85(1):87–94.

Malcolm Ross. 2009. Proto Austronesian verbal morphology: A reappraisal. In *Austronesian historical linguistics and culture history: A festschrift for Robert Blust*. Asia-Pacific Linguistics, College of Asia and the Pacific, The Australian National University.

Sidney Siegel and N. John Castellan. 1988. *Nonparametric statistics for the behavioral sciences*, second edition. McGraw–Hill, Inc.

Stacy Fang-ching Teng. 2008. *A reference grammar of Puyuma, an Austronesian language of Taiwan*. Pacific Linguistics, Research School of Pacific and Asian Studies, The Australian National University.

Stacy Fang-ching Teng. 2009. Case syncretism in Puyuma. *Language and Linguistics*, 10(4):819–844.

Stacy Fang-ching Teng. 2018. *Beinan Yu Yufa Gailun*, 2nd edition. Taiwan Nandao Yu Yuyan Congshu. Council of Indigenous Peoples, New Taipei City.

A Abbreviations used in Glossing

Abbreviation	Meaning
1SG	first person singular
3SG	third person singular
AV	actor voice
CAUS	causative
CV	conveyance voice
DEF	definite
GEN	genitive
INDF	indefinite
LNK	linker
LOC	locative (marker)
LV	locative voice
NAME	personal name
NOM	nominative
OBL	oblique
PASS	passive
PJ	projective
PL	plural
POSS	possessive
PV	patient voice
RED	reduplication
SG	singular
TOP	topic

Table 2: Abbreviations used in glossing. With the exception of NAME, all are based on Teng (2008). Some abbreviations were expanded to make the meaning more transparent.

Automatic Evaluation of Linguistic Validity in Japanese CCG Treebanks

Asa Tomita[1] **Hitomi Yanaka**[2,3] **Daisuke Bekki**[1]
[1]Ochanomizu University [2]The University of Tokyo [3]RIKEN
{tomita.asa, bekki}@is.ocha.ac.jp hyanaka@is.s.u-tokyo.ac.jp

Abstract

In Natural Language Inference, the accuracy of systems based on compositional semantics depends on the quality of syntactic analysis, which in turn relies on linguistically valid training and evaluation data, typically provided by treebanks. However, conventional treebank evaluation metrics focus on data coverage and fail to assess the linguistic validity of syntactic structures. This paper proposes novel evaluation methods to enable automatic and multi-faceted assessment of linguistic validity. We apply these methods to a Japanese treebank based on Combinatory Categorial Grammar and report the evaluation results.

1 Introduction

Natural Language Inference (NLI) is one of the core tasks in natural language processing. Among the various approaches to NLI, research on inference systems based on compositional semantics has experienced steady research progress (Mineshima et al., 2015; Abzianidze and Bos, 2017; Hu et al., 2020). In such systems, obtaining linguistically valid syntactic structures and semantic representations is essential because the inference accuracy is strongly influenced by the outputs of syntactic and semantic analyses, which serve as preprocessing for inference. In particular, syntactic and semantic analyses that contain errors can lead to incorrect inference results. Specifically, Japanese syntactic parsers based on Combinatory Categorial Grammar (CCG) (Steedman, 1996, 2000) achieve high accuracy on standard evaluation datasets (Yoshikawa et al., 2017), although their outputs have also been reported to lack linguistic validity, especially when handling complex constructions such as passives and causatives (Bekki and Yanaka, 2023). This discrepancy arises from the fact that both training and evaluation are conducted on treebanks that contain linguistically invalid analyses.

This issue fundamentally stems from the absence of established approaches to evaluating the linguistic validity of treebanks. While many treebanks have been developed for various languages and formal grammars (Marcus et al., 1993; Hockenmaier and Steedman, 2007; Bos et al., 2009; Boxwell and Brew, 2010; Hockenmaier, 2006), their linguistic validity is yet to be evaluated sufficiently. In many cases, the validity of these resources relies on the assumption that they have been constructed or verified by linguists, but such manual assurances do not offer a quantitative measure of linguistic validity. Instead, there is a growing need for principled methods that can evaluate the linguistic validity of treebanks in a systematic way.

Therefore, this study proposes methods for evaluating the linguistic validity of treebanks from both syntactic and semantic perspectives. We apply these evaluation methods to the Japanese CCG treebank constructed by Tomita et al. (2024), and we report our evaluation results.

2 Construction of the CCG treebank

2.1 Combinatory Categorial Grammar

In syntactic parsing grounded in compositional semantics, sentences are transformed into syntactic structures based on formal grammars. Among these, CCG is characterized by having the weakest generative power among mildly context-sensitive grammars in the Chomsky hierarchy, which makes it particularly well-suited for providing sufficient expressivity to capture the essential syntactic structures of sentences. Treebanks based on CCG (Hockenmaier and Steedman, 2007; Tran and Miyao, 2022) are constructed via automatic conversion from treebanks based on Context Free Grammar (CFG; Marcus et al., 1993) or dependency structures (Nivre et al., 2020), and they are used as training and evaluation data for existing syntactic parsers (Yoshikawa et al., 2017; Tian et al., 2020).

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 74–80
August 28-29, 2025 ©2025 Association for Computational Linguistics

2.2 Linguistically Valid Japanese CCG Treebank

A representative CCG treebank for Japanese is the Japanese CCGbank (Uematsu et al., 2013), which was constructed via automatic conversion from a corpus of dependency structures. It has been widely used as training and evaluation data for Japanese CCG parsers. However, Bekki and Yanaka (2023) pointed out that the Japanese CCGbank contains errors in the analysis of sentences involving case alternations such as passive and causative constructions. To address this issue, Tomita et al. (2024) proposed a new method called *Reforging* (described in Section 2.2.2) that uses the Japanese syntactic parser *lightblue* (Bekki and Kawazoe, 2016) to construct *lightblue CCGbank*, a Japanese CCG treebank designed specifically with a focus on linguistic validity.

2.2.1 Japanese Syntactic Parser *lightblue*

lightblue is a Japanese syntactic parser based on CCG. Its lexicon follows the theoretical analysis of Bekki (2010), particularly in the design of syntactic features, allowing it to generate structures with detailed information such as verb conjugation forms. In contrast, the Japanese CCGbank lacks such feature granularity, making it difficult to constrain syntactic structures—especially for closed-class words.

For open-class words, *lightblue* partially builds its lexicon using lexical information from the morphological analyzer Juman (Kawahara and Kurohashi, 2006). Unlike neural parsers trained on treebanks, it parses without supervision, relying on a precompiled lexicon and CCG combinatory rules. However, inaccuracies in predicate-argument structures remain, limiting the linguistic validity of its analyses.

2.2.2 Overview of Reforging

To address the issue of *lightblue* mentioned above, prior work (Tomita et al., 2024) has introduced a module that integrates argument structures into lexical entries by extracting them from external linguistic resources (Kubota et al., 2020; Ueda et al., 2023) and modifies *lightblue* lexical entries by adding or removing argument structure information. This module in combination with the parser *lightblue* forms the treebank method called *Reforging*. Using this method, the *lightblue CCGbank* was constructed as a linguistically valid Japanese CCG treebank. The dataset consists of 13,653 sentences extracted from ABCTreebank (Kubota et al., 2020), each assigned a CCG syntactic structure and semantic representation based on Dependent Type Semantics (DTS; Bekki and Mineshima (2017)). The remaining challenge is how to evaluate the linguistic validity of the *lightblue CCGbank*.

3 Treebank Evaluation

3.1 Conventional Evaluation Metrics

Treebanks are typically evaluated using metrics such as lexical coverage and parser accuracy. However, in this section, we point out that these conventional evaluation metrics are not comprehensive and are insufficient for evaluating the linguistic validity of treebanks.

3.1.1 Lexical Entries and Coverage Rate

A lexicon can be constructed from the words appearing at the leaf nodes of a parse tree, and metrics such as the number of lexical entries and lexical coverage can be used to evaluate the comprehensiveness of the treebank. Lexical coverage refers to the proportion of words for which the grammar assigns a gold-standard category.

It is important to note that high lexical coverage does not guarantee the linguistic validity of the dataset. Coverage merely indicates how extensively the lexicon can assign some category to encountered words, but it does not evaluate whether the treebank data itself is linguistically valid. Therefore, even a high coverage rate does not ensure the quality or validity of the data.

3.1.2 Parsing Accuracy

In parser-based evaluation, a treebank is used to train the parser, and its accuracy is evaluated by measuring how well it can analyze the syntactic structures of input sentences. Software tools such as evalb[1] are commonly used to compute metrics including precision, recall, F-score, and tagging accuracy.

Although parsing accuracy is commonly used to evaluate syntactic parsers, it does not necessarily reflect the linguistic validity of the underlying dataset. Since accuracy measures alignment with gold-standard annotations, a parser may achieve high scores even when trained on erroneous data. Accordingly, high parsing accuracy alone cannot be taken as evidence of a linguistically valid treebank.

[1] https://nlp.cs.nyu.edu/evalb/

3.2 Evaluation of Linguistic Validity

As discussed in Section 3.1, conventional methods for evaluating treebanks are not sufficient for the quantitative evaluation of linguistic validity. Moreover, evaluating CCG syntactic structures requires advanced knowledge of computational linguistics, making manual evaluation costly and impractical for large-scale treebank validation.

Therefore, this study proposes an automatic method for evaluating the linguistic validity of large-scale Japanese CCG treebanks. In *lightblue CCGbank*, each sentence is assigned a CCG syntactic structure and DTS semantic representation. Building on this data, we introduce two evaluation metrics, one for syntax and one for semantics. By combining these metrics, a multidimensional evaluation approach is achieved.

3.2.1 Syntax-Based Evaluation

Because all sentences in *lightblue CCGbank* are extracted from ABCTreebank, each syntactic structure in the former corresponds to one in the latter. Assuming that ABCTreebank, which was constructed via expert annotation, provides linguistically valid structures, we evaluate the reliability of *lightblue CCGbank* by scoring its alignment with ABCTreebank.

The ABC grammar used in ABCTreebank is a form of categorial grammar that employs function application and composition rules. However, because the definitions of syntactic categories and unary rules differ between ABC grammar and CCG, direct comparison is impossible. To enable comparison, the syntactic categories in ABCTreebank are converted to their CCG counterparts, and alignment is scored based on the following procedure as shown in Figure 1:

1. Convert the ABC grammar into CCG.

2. For each syntactic structure obtained in 1 and its counterpart in *lightblue CCGbank*, create a list of pairs consisting of syntactic categories and phonetic forms.

3. Calculate the score as the proportion of elements in ABCTreebank list that are included in the *lightblue CCGbank* list.

This method has two advantages: one is to compare empty categories in CCG with unary rules in ABCTreebank, and another is to accommodate differences in predicate analysis. However, it also has limitations: it assumes that ABCTreebank is entirely correct, which may not necessarily be the case, and it cannot evaluate syntactic features not annotated in ABCTreebank.

3.2.2 Semantics-Based Evaluation

All syntactic structures in *lightblue CCGbank* are assigned DTS semantic representations. DTS is a proof-theoretic semantic framework based on Dependent Type Theory (DTT; Martin-Löf (1984)). We propose a method for evaluating the validity of DTS semantic representations using type-theoretic verification, known as "type checking". Type checking is a procedure for verifying whether a semantic representation has a well-formed type; if the representation can be proven to have the type `type`, then the check is considered successful.

Type checking fails when the semantic representation is ill-formed. However, it is theoretically proven that when CCG and DTS are used as the syntactic and semantic frameworks, semantic representations should always be well-typed (Bekki, Forthcoming). Therefore, a failure in type checking suggests errors in the implementation of lexical items or combinatory rules that yield ill-typed semantic representations cannot be considered linguistically valid under this system. This property enables the evaluation of syntactic validity from the perspective of semantic compositionality.

A notable strength of this method is that it evaluates syntactic structures at the semantic level based on type theory. However, passing type checking does not necessarily imply linguistic validity of the associated syntactic structures. Thus, syntactic scores and type-theoretic verification serve complementary functions, and their combined use is essential for a comprehensive assessment of treebank quality.

4 Evaluation Experiment

4.1 Experimental Setup

In total, 760 sentences were sampled from various genres within *lightblue CCGbank* and used for evaluation. The syntactic structures were comprehensively evaluated based on the following metrics.

Syntactic Structure Score Average Using the method in Section 3.2.1, each sentence was scored by the percentage of matching `(surface form, syntactic category)` pairs, and averages were calculated per genre.

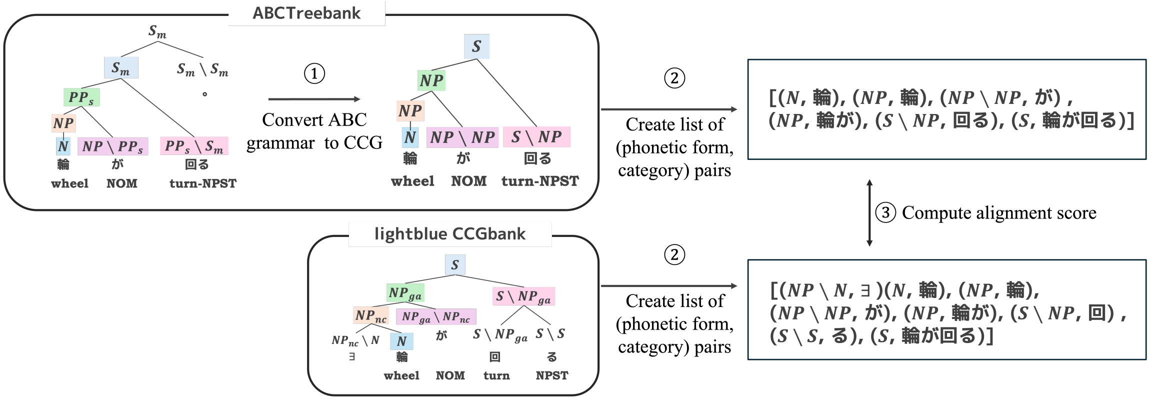

ABCTreebank
S_m
S_m
$S_m \backslash S_m$
。
PP_s
NP
N
$NP \backslash PP_s$
$PP_s \backslash S_m$
輪
が
回る
wheel
NOM
turn-NPST
①
Convert ABC grammar to CCG
S
NP
NP
N
$NP \backslash NP$
$S \backslash NP$
輪
が
回る
wheel
NOM
turn-NPST
②
Create list of (phonetic form, category) pairs
[(N, 輪), (NP, 輪), (NP \ NP, が) , (NP, 輪が), (S \ NP, 回る), (S, 輪が回る)]
③ Compute alignment score
lightblue CCGbank
S
NP_{ga}
$S \backslash NP_{ga}$
NP_{nc}
$NP_{ga} \backslash NP_{nc}$
$NP_{nc} \backslash N$
N
が
$S \backslash NP_{ga}$
$S \backslash S$
ヨ
輪
回
る
wheel
NOM
turn
NPST
②
Create list of (phonetic form, category) pairs
[(NP \ N, ヨ)(N, 輪), (NP, 輪), (NP \ NP, が), (NP, 輪が), (S \ NP, 回) , (S \ S, る), (S, 輪が回る)]

Figure 1: Scoring Process in Syntactic Evaluation

Genre	Number of Data	Average Score	Type Checking Pass Rate	Overall Score
Aozora Bunko	125	42.4	63.2	20.89
Bible	40	49.1	57.5	21.57
Books	10	49.8	60.0	33.33
Dictionary	100	55.59	57.0	28.57
Proceedings	35	41.8	77.1	23.91
Fiction	30	51.1	66.7	31.82
Law	10	33.4	80.0	28.57
Other	50	50.2	64.0	26.47
News	50	40.4	78.0	21.88
Non-fiction	10	53.4	100.0	33.33
Spoken Language	50	36.98	88.0	25.37
TED Talks	25	41.68	64.0	21.88
Textbooks	200	49.59	60.0	27.54
Wikipedia	25	45.88	88.0	32.43
Total	760	46.0	66.2	26.00

Table 1: Evaluation Results

Type Checking Passage Rate Based on Section 3.2.2, type checking was performed on the semantic representations. The passage rate was defined as the proportion of sentences with semantic representations successfully verified as well-typed.

Overall Evaluation This metric is the percentage of sentences with a syntactic score of 50.0 or higher that also passed type checking, indicating both syntactic and semantic validity.

4.2 Results and Discussion

The results of the experiment are presented in Table 1. The overall average syntactic structure score was 46.0, with 503 out of the 760 evaluated sentences passing type checking, yielding a pass rate of 66.2%.

An important finding is that there is no clear correlation between the average syntactic structure score average and the type checking passage rate, suggesting that the two metrics capture orthogonal properties relevant to linguistic validity. For instance, in the law genre, although the type checking passage rate is as high as 88%, the syntactic score remains relatively low at 33.4%, indicating that syntactic alignment and semantic well-formedness are independent. This observation highlights the complementary nature of the two evaluation metrics. Even if a parse tree receives a high syntactic score, indicating structural similarity to gold-standard annotations, it cannot be regarded as linguistically valid if it fails type checking. Passing the type checking procedure serves as a necessary condition for semantic consistency, verifying that the semantic composition is correctly implemented within the DTS framework.

4.3 Manual Evaluation

To assess the reliability of our syntax-based evaluation metric, we compared its results against a manually annotated subset of 152 sentences from the lightblue CCGbank. The results are shown in Table 2.

The metric achieved a precision of 0.64, recall of 0.79, F1-score of 0.71, and accuracy of 0.74 with respect to human judgments. These results suggest that the syntax-based evaluation has relatively high recall, meaning it is capable of capturing most linguistically valid structures identified by human annotators. However, the lower precision indicates that some sentences deemed valid by the metric may not align with human judgments, possibly due to overpermissive category matching. Overall, the moderate F1-score (0.71) and reasonably high accuracy (0.74) indicate that the syntax-based metric can serve as a useful proxy for linguistic validity, though it may require further refinement to reduce false positives.

		Manual Evaluation	
		True	False
Score > 50	True	48	27
	False	13	64
Accuracy		0.739	
Precision		0.640	
Recall		0.787	
F1		0.706	

Table 2: Confusion Matrix and Manual Evaluation Results

5 Limitations and Future Work

5.1 Annotation Errors in the ABCTreebank

Although the average syntactic alignment score appears relatively low at 46%, this result is partially attributable to annotation errors in the ABCTreebank, which serves as the gold standard in our evaluation. Our evaluation assumes that ABCTreebank provides linguistically valid structures; hence, any inaccuracies in its annotations directly affect the computed scores.

For instance, determiners are annotated as N/N, a category that yields a noun. However, they should more appropriately be labeled as NP/N, since they functionally yield noun phrases. Such inconsistencies in category assignment can reduce alignment scores, even when the underlying syntactic structures are otherwise linguistically sound.

5.2 Limits of Cross-Framework Evaluation

Some category mismatches observed in our evaluation — such as annotating determiners as N/N in ABCTreebank, while they are assigned NP/N in lightblue CCGbank — might appear to be minor inconsistencies. However, such differences are not simply attributable to the annotation rules; rather, they reflect deeper theoretical assumptions about the treatment of syntactic categories. In CCG, for example, NP/N indicates that a determiner produces a complete noun phrase, aligning with its semantic interpretation and compositional properties. In contrast, frameworks like ABC grammar often avoid using NP entirely, opting for a more uniform treatment of nouns and noun phrases.

This highlights a broader challenge for our evaluation method; it is not simply a conversion from one formal description to another, but a translation between distinct linguistic theories. Consequently, it necessitates a careful alignment of theoretical assumptions across frameworks. Each theory prioritizes different linguistic principles. Without explicitly addressing these theoretical discrepancies, evaluation scores may primarily reflect inter-framework divergences rather than actual linguistic inaccuracies. In other words, a mismatch between NP/N and N/N might not indicate a parsing error, but rather a fundamental theoretical difference in how the grammar encodes syntactic categories.

5.3 Future Work

While our evaluation is currently conducted within the CCG and DTS frameworks, the proposed metrics are designed to be framework-agnostic. Future work will involve investigating their applicability to other syntactic and semantic frameworks, such as CFG and Abstract Meaning Representations (Langkilde and Knight, 1998), thereby further substantiating the generality of our evaluation method. Moreover, we intend to enhance the validity of the *lightblue CCGbank* through the incorporation of feedback mechanisms into the treebank construction process.

6 Conclusion

This study proposed syntactic and semantic evaluation metrics for assessing the linguistic validity of treebanks from two independent perspectives. These metrics enable a more fine-grained analysis of structural validity than conventional approaches. Ensuring the validity of treebanks is essential not only for improving inference accuracy but also for satisfying requirements such as transparency of error detection and enhanced explainability in future language processing systems. By addressing the lack of principled evaluation methods for linguistic validity, this work offers a step toward more reliable and linguistically grounded approaches in NLP.

Acknowledgments

This work was supported by JST BOOST, Japan Grant Number JPMJBS2406, JSPS KAKENHI Grant Number JP23H03452, Japan, and JST CREST Grant Number JPMJCR20D2, Japan.

References

Lasha Abzianidze and Johan Bos. 2017. Towards universal semantic tagging. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS) — Short papers*.

Daisuke Bekki. 2010. *Nihongo-Bunpoo-no Keisiki-Riron - Katuyootaikei, Toogohantyuu, Imigoosei - (trans. 'Formal Japanese Grammar: the conjugation system, categorial syntax, and compositional semantics')*. Kuroshio Publisher, Tokyo.

Daisuke Bekki. Forthcoming. From Dependent Type Theory to natural language semantics.

Daisuke Bekki and Ai Kawazoe. 2016. Implementing variable vectors in a CCG parser. In *Logical Aspects of Computational Linguistics. Celebrating 20 Years of LACL (1996–2016)*, pages 52–67, Berlin, Heidelberg. Springer Berlin Heidelberg.

Daisuke Bekki and Koji Mineshima. 2017. *Context-Passing and Underspecification in Dependent Type Semantics*, pages 11–41. Springer International Publishing, Cham.

Daisuke Bekki and Hitomi Yanaka. 2023. Is Japanese CCGBank empirically correct? a case study of passive and causative constructions. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 32–36, Washington, D.C. Association for Computational Linguistics.

Johan Bos, Cristina Bosco, and Alessandro Mazzei. 2009. Converting a dependency treebank to a categorial grammar treebank for Italian. In *Proceedings of the eighth international workshop on treebanks and linguistictheories (TLT8)*, pages 27–38, Italy, Milan.

Stephen A. Boxwell and Chris Brew. 2010. A pilot Arabic CCGbank. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Julia Hockenmaier. 2006. Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 505–512, Sydney, Australia. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kuebler. 2020. MonaLog: a lightweight system for natural language inference based on monotonicity. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 334–344, New York, New York. Association for Computational Linguistics.

Daisuke Kawahara and Sadao Kurohashi. 2006. Case frame compilation from the web using high-performance computing. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 1344–1347, Genoa, Italy. European Language Resources Association (ELRA).

Yusuke Kubota, Koji Mineshima, Noritsugu Hayashi, and Shinya Okano. 2020. Development of a general-purpose categorial grammar treebank. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5195–5201, Marseille, France. European Language Resources Association.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 704–710, Montreal, Quebec, Canada. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Per Martin-Löf. 1984. *Intuitionistic Type Theory Vol. 1*. Bibliopolis.

Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061, Lisbon, Portugal. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Mark Steedman. 1996. *Surface Structure and Interpretation*. The MIT Press, Cambridge.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

Yuanhe Tian, Yan Song, and Fei Xia. 2020. Supertagging Combinatory Categorial Grammar with attentive graph convolutional networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6037–6044, Online. Association for Computational Linguistics.

Asa Tomita, Hitomi Yanaka, and Daisuke Bekki. 2024. Reforging : A method for constructing a linguistically valid Japanese CCG treebank. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 196–207, St. Julian's, Malta. Association for Computational Linguistics.

Tu-Anh Tran and Yusuke Miyao. 2022. Development of a multilingual CCG treebank via Universal Dependencies conversion. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5220–5233, Marseille, France. European Language Resources Association.

Nobuhiro Ueda, Kazumasa Omura, Takashi Kodama, Hirokazu Kiyomaru, Yugo Murawaki, Daisuke Kawahara, and Sadao Kurohashi. 2023. KWJA: A unified Japanese analyzer based on foundation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 538–548, Toronto, Canada.

Sumire Uematsu, Takuya Matsuzaki, Hiroki Hanaoka, Yusuke Miyao, and Hideki Mima. 2013. Integrating multiple dependency corpora for inducing wide-coverage Japanese CCG resources. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1042–1051, Sofia, Bulgaria. Association for Computational Linguistics.

Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. A* CCG parsing with a supertag and dependency factored model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 277–287, Vancouver, Canada. Association for Computational Linguistics.

Metaphorical Heads and Literal Dependents: Syntactic Properties of Metaphors in German

Stefanie Dipper
Sprachwissenschaftliches Institut / CRC Metaphors of Religion
Fakultät für Philologie
Ruhr-Universität Bochum
stefanie.dipper@rub.de

Abstract

In this paper we examine the way metaphors are expressed in language. Our starting hypothesis is that the two expressions that are central to metaphor – namely the metaphorical expression and the expression that represents the target of the metaphorical transfer – typically stand in a syntactic dependency relation: metaphorical heads govern literal dependents. An analysis of German sermons with 30k words confirms that the hypothesis applies in 67% of the cases. 10% show the reverse relationship and in 23% there is a common ancestor.

1 Introduction

According to Lakoff and Johnson (1980), metaphor is a basic cognitive phenomenon in which a (simpler) concept is used to make another (more complex) concept comprehensible. The two concepts come from different domains, typically the simpler concept is more concrete, more physical, and the more complex concept is more abstract. In our work, we are interested in the way metaphors are expressed in language (and we refer to metaphorically-used expressions also as "metaphors").

The transfer from one domain to another results in a semantic "clash" typical of metaphors, in which semantically (actually) incompatible expressions collide. (1) and (2) are examples of this: *Völker* 'peoples' is normally not a possible attribute of *Meer* 'sea' (see the highlighted expressions in the English translation). Similarly, an (abstract) soul cannot be nourished.

(1) *jedes Gesicht im Meer der Völker wird uns erfreuen wie das Gesicht eines Freundes* (P4150, s48)[1]
'every face <u>in the sea of peoples</u> will delight us like the face of a friend'

(2) *wie uns Gottes Güte ganz nah kommt und unsere Seele satt wird* (P5714, s80)
'how God's goodness comes very close to us and <u>our soul is full (nourished)</u>'

Following the terminology of Lakoff and Johnson (1980), we refer to the semantic domain of the metaphorical, transferred expression (*sea; full*) as the source domain and the semantic domain into which it is transferred (i.e. the domains of *peoples; soul*) as the target domain. We call the corresponding expressions source-domain and target-domain expressions, or source and target expressions for short.

If a source expression is transferred, this means that suitable properties of it are transferred to the target expression. For example, in (1) this could be the property of the sea to be composed of countless individual parts (the waves) that are difficult to distinguish. 'Every face in the sea of peoples' could then mean: 'Every face in the anonymous, indistinguishable mass of peoples'.

The transferred property is called *tertium comparationis*, which is sometimes explicitly mentioned in the text, but often has to be inferred by the listener. (1) could be reformulated as follows to make the tertium comparationis explicit: The peoples are like the sea, insofar as individual faces are as indistinguishable as individual waves.

This paper examines the syntactic properties of metaphors in German. The focus is on the expressions that cause the semantic clash and thus trigger the metaphorical interpretation. Our hypothesis is that these expressions typically stand in specific syntactic relations: typically, the source-domain expression is the syntactic head of the target-domain expression, which functions as the dependent.

This paper investigates this hypothesis using German data from sermons that has been automatically

[1] Source: sermon ID = P4150; sentence ID = s48 from https://gitlab.ruhr-uni-bochum.de/comphist/predigtenkorpus, (Dipper and Roussel, 2024).

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 81–90
August 28-29, 2025 ©2025 Association for Computational Linguistics

enriched with Universal Dependencies and additionally manually annotated for metaphors.

2 Related work

In the framework of Construction Grammar (CxG), Croft (1993) and especially Sullivan (2009) investigated relations between source and target expressions. The generalization here is that source expressions tend to be conceptually-dependent elements, while the target expressions are conceptually-autonomous elements – i.e. the dependencies would be exactly the opposite of what I claim. However, "dependency" is defined semantically in this framework and thus differently than in syntax: Autonomous elements are those that can be conceptualized on their own. The meaning of dependent elements, on the other hand, depends on the conceptualization of another element. Thus, for example, verbs are typically (semantically) dependent on their subject, which plays the role of an (obligatory) agent or actor, as in (3) (from Sullivan, 2009). In any syntactic analysis, however, the relationship would be reversed: the verb would be the head and the subject the dependent.

(3) *The cinema beckoned.*

In the CxG framework, one and the same syntactic construction can come with different semantic dependencies. For instance, adjective–noun constructions can be instances of "predicating modifier constructions" (as in (4a)) or "domain constructions" (as in (4b)). In (4a) the adjective is considered dependent (and metaphorical), in (4b) the adjective is considered autonomous (and literal).

(4) a. *bitter thoughts*
b. *political game*

To our knowledge, the question of the *syntactic* relation between the source and the target expression has not yet been systematically investigated. However, it is noticeable that many works on the analysis of metaphors focus on typical head–dependent relations, in particular verb–subject (V+S), verb–object (V+O), adjective–noun (A+N) and subject–predicative (S+Pred) pairs (i.e. copula constructions). (5) shows examples of these relations (from Krishnakumaran and Zhu, 2007). Table 1 lists a number of relevant works and the metaphor structures examined in them.

(5) a. V+O: *He planted good ideas in their minds.*
b. A+N: *He has a fertile imagination.*
c. S+Pred: *He is a brave lion.*

Table 1 also shows that some studies are limited to pre-selected lemmas. For example, Turney et al. (2011) selected a total of five adjectives for the A+N pairs (*dark, deep, hard, sweet, warm*), which were combined with typical nouns, such as *dark glasses* or *dark mood*.

In pairings with verbs, it is the verb which is potentially metaphorical in general and the subject or object is literal. In the A+N pairs, the adjective is usually the metaphor candidate and the noun is literal (as in example (4a)). This is probably due to the fact that the adjectives were often carefully selected to be concrete, perceptible qualities and, hence, to have clear metaphorical potential.

3 Corpus

3.1 Data

For this study, we use 15 randomly-selected Protestant German-language sermons with almost 30k tokens. The sermons come from a corpus compiled by Dipper and Roussel (2024). They are based on different Bible texts and some of the sermons have been created for special occasions such as baptisms or golden confirmations.

We automatically enriched the sermons with lemmas, word types (STTS and UPOS) and dependencies according to UD (see Section 3.2 for details).[2]

We also manually annotated the sermons with information on metaphors (see Section 3.3 for details of the annotations). The curated versions form the basis for the analysis in this paper.[3]

Table 2 lists the basic statistics for the analyzed data. The sermons differ greatly in terms of overall length (1,249–3,121 tokens). This is partly due to the fact that the Bible text discussed in the sermon is only sometimes included in the sermon itself. But the length of the sentences also varies greatly, from 1–105. A typical sentence of length 1 is the word "Amen" at the end of the sermon.

3.2 Universal Dependencies

Universal Dependencies (UD, de Marneffe et al., 2021) is a universally applicable framework for

[2] Lemma, STTS, UPOS: Stanza with model de_gsd_charlm from Qi et al. (2020); dependency relations: 3_mhg_modg from Haiber (2024).

[3] The data is made freely available at https://gitlab.ruhr-uni-bochum.de/comphist/syntaxfest2025_metaphors.

Study	Structures considered	Lemmas considered
Krishnakumaran and Zhu (2007)	S+Pred; V+O; A+N	all
Turney et al. (2011)	V; A+N	A, V: selection
Shutova et al. (2013)	V+S; V+O	all
Gandy et al. (2013)	S+Pred; V+O; A+N	N: selection
Tsvetkov et al. (2014)	V; A+N	V: selection (EN data)
Bizzoni et al. (2017)	A+N	A: selection

Table 1: Syntactic structures of metaphors examined in the respective studies (V: verb; S: subject; O: object; Pred: predicative; A: adjective; N: noun). The “Lemmas considered” column indicates whether only a selection of predefined lemmas were taken into account.

Statistics	Mean	SD	Min	Max
#tok/doc	1,989.7	450.9	1,249	3,121
#sent/doc	114.3	26.7	83	188
#tok/sent	17.4	12.6	1	105

Table 2: Basic statistics about the corpus: mean, standard deviation, minimum and maximum of different measures. The total number of tokens is 29,846.

syntax analysis in the form of dependency relations. These relations exist between two words, one of which is the head and the other the dependent. The relations are labeled with the syntactic function of the dependent in relation to the head, e.g. a noun can act as the subject (`nsubj`) of a verb head or an adjective acts as the modifier (`amod`) of a noun head. A special feature of UD is that content words (and not function words) are the heads, as function words are less universal than content words and depend rather heavily on the individual language.

The UD principles are spelled out on a language-specific basis. For our analysis, we use the scheme for German presented in Dipper et al. (2024a).

For many relations (such as verb–argument relations, head–modifier relations), the hierarchical structure of the dependencies is undisputed. An exception are copula(-like) constructions of the form *A is B* – which is the canonical form of stating conceptual metaphors, such as ARGUMENT IS WAR (Lakoff and Johnson, 1980). According to the official UD guidelines, only certain lemmas (e.g. English *be* or German *sein* ‘be’) may function as the verb in a copula construction, and the predicative is analyzed as the head of the construction. Constructions with the lemma *become* (as in *A becomes B*) are analyzed differently: here the verb is the head and the predicative is an argument of it. In Dipper et al. (2024a), however, German *werden* ‘become’ is also analyzed as a copula verb, since copula *sein* and *werden* share many grammatical properties.

Since we follow Dipper et al. (2024a) in our data, we analyze the adjective *satt* ‘full’ as a predicative and head in example (2), which of course influences the evaluation of head–dependency relations.

3.3 Annotations

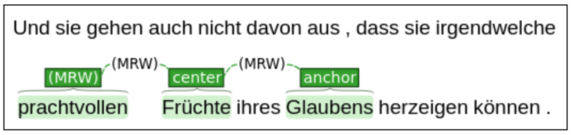
The majority of metaphors in language are conventionalized and typically do not stand out at all. We are particularly interested in a subset of metaphors: *deliberate metaphors* in the sense of Steen (2008). These are metaphors which are used *as* metaphors and of which speakers and listeners are likely aware.

In Dipper et al. (2024b), guidelines for the annotation of deliberate metaphors are presented.[4] We use a slightly modified version of these guidelines, which are briefly described below.

Deliberate metaphors are a rather common phenomenon in sermons. This is one of the reasons why we annotate sermons.

Deliberate metaphor We annotate all deliberate metaphors in the complete sermon. For this purpose, all metaphorical expressions are labeled and all expressions within a clause that belong to the same metaphorical image are additionally linked to each other so that they form a chain. The labels distinguish between expressions that are central to the metaphorical image (labeled as `center`) and those that are less central (`MRW` for “metaphor-related

[4] These guidelines were evaluated on four German TEDx talks, using the γ agreement measure (Mathet et al., 2015), with γ scores of 0.35, 0.43, 0.49 and 0.56. For details, see Dipper et al. (2024b).



Und sie gehen auch nicht davon aus , dass sie irgendwelche prachtvollen Früchte ihres Glaubens herzeigen können .

Figure 1: Annotation of a metaphor, with center, MRW and anchor, see example (6) (screenshot of INCEpTION).

word", after Steen et al., 2010). The chain also includes expressions of the target level to which the metaphorical transfer takes place and which participate centrally in the semantic clash between source and target expressions. The target expressions are annotated with the label `anchor`.

Figure 1 shows the annotation of (6) as an example. The annotation was created with the tool INCEpTION (Klie et al., 2018). The corresponding biblical text is the parable of the vine, which is often referred to in that sermon. In the example, *Früchte* 'fruits' is annotated as the `center`, as they are meant metaphorically as a positive yield of faith. The adjective *prachtvoll* 'gorgeous' belongs to the image of fruit and is therefore annotated as `MRW`. The semantic clash occurs between *Früchte* 'fruits' and *Glauben* 'faith', since the abstract faith cannot produce real fruits.[5] The edges of the chain are labeled with `MRW` to distinguish them from other chain types.[6]

(6) *Und sie gehen auch nicht davon aus, dass sie irgendwelche prachtvollen Früchte ihres Glaubens herzeigen können.* (P5634, s101)
'Nor do they [= certain Christians] assume that they can show off any magnificent fruits of their faith.'

An important difference between the annotation of source and target expressions is that we annotate all source expressions, while from the target domain we only annotate those expressions that are directly affected by the metaphor, e.g. because they are the target of the metaphorical transfer. Of these target expressions, we only annotate the head.[7] This means that the number of expressions annotated as metaphorical cannot be meaningfully compared directly with the number of anchor expressions.

The chains are usually restricted to a clause, since we are interested in syntactic relations and these are restricted to clauses.[8] However, metaphorical images often extend over several subordinate clauses. In many cases, though, the link between the clause-internal chains is established in the text itself by anaphoric reference, as in (7), see Figure 2 for the annotations. We encode anaphoric relations by means of coreference links. In Figure 2, the first chain is <opens-up, understanding> (marked in brown), the second chain is <that [= understanding] + us + carry> (in green). The coreference link connects the antecedent *Verstehen* 'understanding' with the relative pronoun *das* 'that' and is also marked in green, with the labels `(Coref)`; this label is also used for the chain edges.

(7) *In manchen Momenten öffnet sich ein tiefes Verstehen, das uns tragen kann und eine Hoffnung gibt, über die Stunde und über den Tag hinaus.* (P5151, s53)
'In some moments, a deep understanding opens up that can carry us and gives us hope beyond the hour and beyond the day.'

Note that the part *und eine Hoffnung gibt* 'and gives us hope' is not annotated in Figure (2). The reason for this is that *geben* 'give' is a semantically faded verb and the metaphor of giving is strongly conventionalized if not lexicalized here.

Comparisons In addition to canonical metaphors, there are also comparisons in which properties of one expression are transferred to another expression in a similar way to metaphors. However, comparisons are signaled overtly, usually by the expression *wie* 'as'.

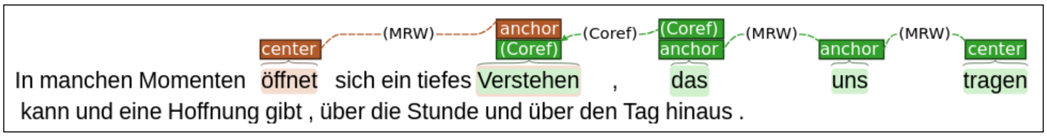
As a consequence, the syntactic structure changes so that the semantic clash no longer occurs between verb and object or adjective and noun, but between a target expression and the *as*-phrase.

[5] It would also be conceivable to include the verb *herzeigen* 'show off' in the chain as `MRW`. The annotators have decided against this, possibly because the base verb *zeigen* 'show' can be regarded as semantically faded.

[6] We use default labels both for the non-central MRW expressions and for the edge labels. Such default labels are displayed in brackets in INCEpTION: `(MRW)`.

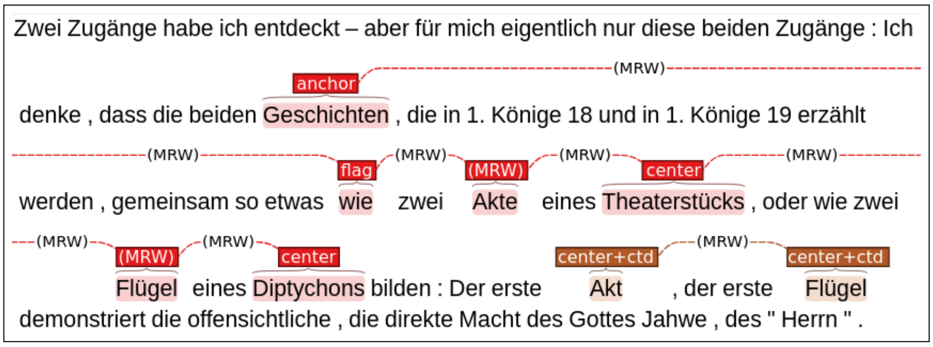
[7] By default, all expressions that are not annotated as metaphorical belong to the target domain. Therefore, we restrict the annotation of target expressions to those expressions that trigger the semantic clash.

[8] There are of course syntactic relations between whole clauses and a head, e.g. between a relative clause and its antecedent or between an adverbial clause and the governing verb. The point here is that there are usually no syntactic relations between a word in an embedded clause and a word in the superordinate clause.



In manchen Momenten öffnet sich ein tiefes Verstehen , das uns tragen kann und eine Hoffnung gibt , über die Stunde und über den Tag hinaus .

Figure 2: Annotation of a metaphor involving a coreference link, see example (7).



Zwei Zugänge habe ich entdeckt – aber für mich eigentlich nur diese beiden Zugänge : Ich denke , dass die beiden Geschichten , die in 1. Könige 18 und in 1. Könige 19 erzählt werden , gemeinsam so etwas wie zwei Akte eines Theaterstücks , oder wie zwei Flügel eines Diptychons bilden : Der erste Akt , der erste Flügel demonstriert die offensichtliche , die direkte Macht des Gottes Jahwe , des " Herrn " .

Figure 3: Annotation of a comparison, see example (8).

This phrase can be of varying complexity, either a simple prepositional phrase or a complex comparison phrase. The relation between target and source expression is typically less close in such cases.

For an example, see (8), which is about two consecutive biblical stories, both of which are about the prophet Elijah and are very different. These two stories are compared to two acts of a play or two wings of a diptych. Figure 3 shows the annotation for the example. The expression signaling the comparison is annotated as `flag`.

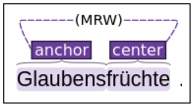
(8) *Zwei Zugänge habe ich entdeckt – aber für mich eigentlich nur diese beiden Zugänge: Ich denke, dass die beiden Geschichten, die in 1. Könige 18 und in 1. Könige 19 erzählt werden, gemeinsam so etwas wie zwei Akte eines Theaterstücks, oder wie zwei Flügel eines Diptychons bilden: Der erste Akt, der erste Flügel demonstriert die offensichtliche, die direkte Macht des Gottes Jahwe, des "Herrn".* (P5157, s17)
'I have discovered two approaches – but for me actually only these two approaches: I think that the two stories told in 1 Kings 18 and 1 Kings 19 together form something like two acts of a play, or like two wings of a diptych: The first act, the first wing demonstrates the obvious, the direct power of Yahweh God, the "Lord."'

In addition to the metaphorical labels already mentioned, there are further labels for special cases. If a metaphorical image is continued over several clauses, it can happen that there are clauses without semantic incongruities. There is a separate label `center+ctd` for this (for 'continued') to indicate that the chain does not contain an `anchor`. Figure 3 shows an example in the second part of the sentence).

Another special case is when an actually conventionalized metaphor is taken up and expanded and thus "revived". For this we use the label `center+rvt` (for 'revitalized'). Cases of doubt that lie between conventionalized and metaphorical use are annotated with `grey_area`.

Finally, in German compounds the semantic clash can take place within a word, cf. example (9). The source-domain part of the compound is its head *Früchte* 'fruits', which is modified by the target-domain expression *Glauben* 'faith'.

(9) *Glaubensfrüchte* (P5634, s61)
faith-fruits
'fruits of faith'



Glaubensfrüchte .

Even if chains are usually restricted to simple clauses, longer chains often result, e.g. through coordination. In the example (8) (Figure 3) there is such a coordination: 'two acts of a play' is coordinated with 'two wings of a diptych'. Such cases are annotated in a common chain.

The annotation in Figure 3 also shows that it is not always possible to make a strict distinction between the different subtypes of metaphorical labels. For example, in the figure, 'play' and 'diptych' are annotated as center. One could just as well have chosen ‘acts’ and ‘wings’ as center if one considers these terms to be more central or more descriptive for the metaphorical image.

In the following analysis, we use the labels MRW for all metaphorical labels, CNT for all center labels, and ANC for the anchor labels.[9] The analysis differentiates between “real” metaphors, which we simply call ‘metaphors’, and explicit comparisons, which we call ‘comparisons’.

4 Results

The difference between (“real”) metaphors and (explicit) comparisons described in Section 3.3 allows us to examine the syntactic properties of both types separately. That is, the comparisons can be treated as a kind of control group for which we expect different syntactic relations than for the metaphors. However, there are also instances in the corpus where both forms have been annotated within the same chain – such annotations are problematic for our analysis. A clear distinction between the two variants is only possible on the basis of the label flag. Mixed chains are then incorrectly assigned to the comparison variant.

However, comparisons are much rarer in the sermons than metaphors. One reason for this is that in longer passages of comparisons we have only marked the whole block as a comparison, without further internal analysis because in such extended comparisons, the metaphorical transfer applies to discourse units rather than at the sentence level, and syntactic relations do not play a role.

We nevertheless include the rare (short) chains with explicit comparisons in the analysis of syntactic properties and dependencies.

We start with an overview of the distribution of the metaphor labels. Then we take a closer look at the parts of speech and grammatical functions involved. Finally, we look at the dependency paths between source and target expressions.

Metaphor labels In total, there are 1,029 annotations (374 chains) for metaphors and 80 annotations (27 chains) for comparisons in the 15 sermons. Table 3 shows the distribution averaged across the individual documents. The standard deviation is very high, one sermon contains a total of only 13 metaphor annotations, another 142. The differences are of course (partly) dependent on the length of the sermons. The table therefore also shows the statistics on the percentage distributions, calculated against the total number of tokens in a document. Here, too, there is a large discrepancy between the extremes: a sermon with 0.6% metaphor annotations vs. one with 7.7%. Comparisons occur very rarely overall. This skewed distribution is also reflected in the number of chains. The chain lengths (in number of chain members) mainly range between 2 and 4. Comparison chains are somewhat longer than metaphor chains.

	Mean	SD	Min	Max
#Annotations				
Metaphors	68.6	41.1	13	142
Comparisons	7.3	4.3	3	17
Metaphors (%)	3.5	1.9	0.6	7.7
Comparisons (%)	0.3	0.1	0.1	0.6
#Chains				
Metaphors	24.9	14.4	5	51
Comparisons	2.5	1.5	1	6
Chain length				
Metaphors	2.8	1.0	1	7
Comparisons	3.0	1.1	1	6

Table 3: Average number of annotations and chains and average chain lengths of metaphors and comparisons in all documents. The target expressions (anchors) are included in the counts.

Figure 4 visualizes the (percentage) distribution of MRW and ANC labels (i.e. source vs. target expressions) in the individual documents, sorted according to the percentage frequency of MRW labels.

For the most part, the number of MRW vs. ANC is roughly comparable. However, the sermons P5634 and P5354 have a significantly higher proportion of metaphors. In P5634, for example, this is due to the fact that the image of the vine is treated and developed very prominently in the sermon (cf. example (6)), e.g. the lemma *Weinstock* ’vine’ appears a total of 4 times and *Frucht* ‘fruit’ 3 times as a metaphor.

Conversely, anchors predominate in other sermons. Typical target expressions in the sermons are

[9] We consider the following labels as CNT: center, center+rvt, grey_area, anchor|center* (i.e. center and its variants in compounds). The label ANC subsumes the labels anchor and center*|anchor (in compounds). The label MRW subsumes CNT plus the default label (MRW).

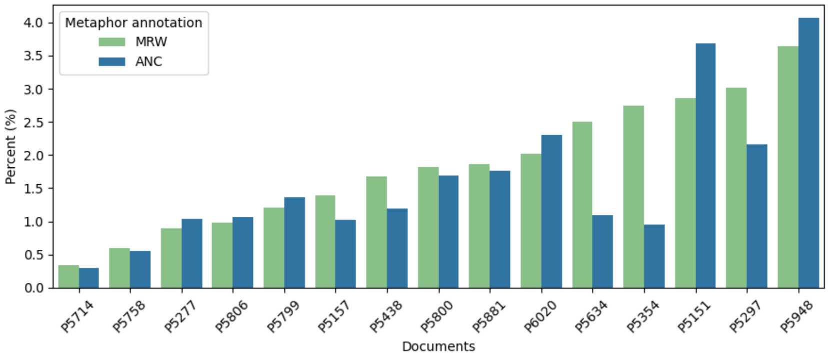


Figure 4: Metaphor (MRW) vs. anchor (ANC) annotations of all documents, sorted according to MRW percentages ("real" metaphors only, ignoring comparisons).

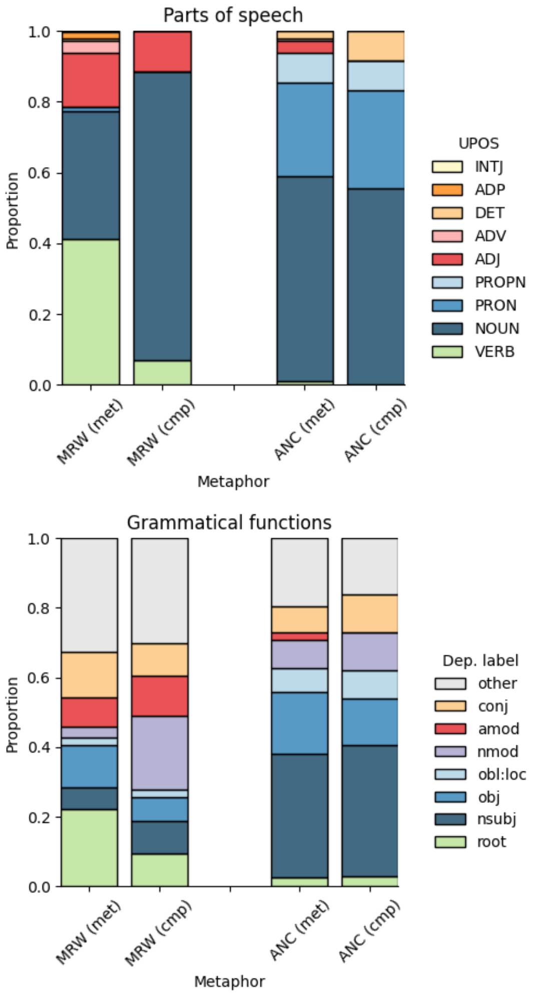


Figure 5: Part-of-speech tags (top) and grammatical functions (bottom) of MRW vs. ANC (left vs. right pairs) among metaphors ('met', first bar) and comparisons ('cmp', second bar). The bars show the proportions of the UPOS and the most frequent dependency labels.

religious expressions such as *Gott* 'god' or *Glaube* 'faith'. Sermon P5151, for example, where anchors clearly predominate, is about the search for God. In a total of 8 metaphorical chains, *Gott* is the target expression.

Parts of speech and grammatical functions The top part of Figure 5 compares the distribution of parts of speech (according to the UPOS tagset), on the one hand between MRW and ANC, and on the other between metaphors and comparisons. One can clearly see that the anchors (right pair of bars) of both types are very similar: in both cases, the bluish noun-like tags (nouns, pronouns, proper names) clearly predominate, whereas verbs are virtually absent.

The situation is clearly different for metaphors (left pair of bars): here, verbs form the largest group with 41.2%, followed by nouns (36.1%) and adjectives (15.4%). However, the (few) comparisons – there are only 43 instances of MRW in comparisons in total – show a distribution that is rather similar to that of the anchors in that nominal categories predominate. Interestingly, there are virtually no pronouns among the metaphors.

This distribution is reflected at the level of grammatical functions (i.e. dependency labels): Figure 5 (bottom part) shows the distribution of labels that were among the 5 most frequent labels of an MRW or ANC. The anchors show typical nominal functions (nsubj, obj, obl:loc, nmod), whereas the metaphors also show verbal functions (root) in addition to nominal functions. The function amod, which is also quite frequent, marks attributive adjectives. Conjuncts (conj, within coordinations) are also found quite frequently in all distributions.

Of course, the two distributions – parts of speech

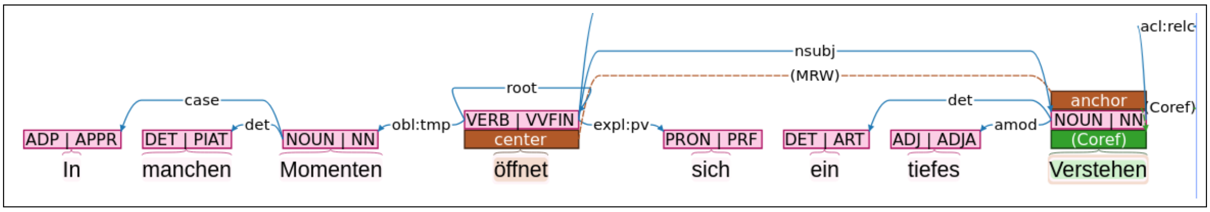


Figure 6: The dependency path `CNT > nsubj > ANC` between the center *öffnet* 'opens' and the anchor *Verstehen* 'understanding', see example (7) and also Figure 2.

and functions – are not independent of each other: a verb, for example, cannot function as `nsubj`.

Nevertheless, these findings provide initial support for the hypothesis that the metaphor is typically the syntactic head of its anchor: `root` is the function of the root of a sentence, i.e. the highest node of all. Functions like `nsubj`, `obj`, `obl:loc`, `nmod` are functions that depend on other words (`nsubj` on verbs or predicatives; `obj` and `obl:loc` on verbs; `nmod` on nouns).

Dependency paths Next, we examine what kind of dependency path exists between a center and its anchor. To do this, we only look at relations within one chain. The algorithm for determining the path between two words in a chain is as follows:

1. First calculate the paths to the root for both words. In these paths, skip `conj` relations (as they are only technically required in UD in order to connect several conjuncts).

2. If one of these paths is empty, this word is the root and dominates the other word. The dependency path consists of the non-empty path.

3. If both paths are not empty, but one of the paths is a real sub-path of the other, the subpath is subtracted from the longer path. The word with the shorter path dominates the other via the remaining dependency path.

4. If both paths are not empty and neither is a true sub-path of the other, the system searches for overlaps between the two paths (i.e. nodes that dominate both words and can therefore be ignored). Intersections are deleted, the concatenated remaining paths form the dependency path between the two words. None of the words dominates the other.

Since there can be several `CNT` and `ANC` within a chain, we first calculate all paths pairwise between all centers and all anchors of a chain, and choose the shortest of these paths as the final path. If several equally short paths exist, we extract all of them.

In total, we extracted 481 dependency paths for metaphors and 26 paths for comparisons.[10] Table 4 shows the paths that occur most frequently. The paths are read as follows: a path starts at a center (`CNT`) and leads to the anchor (`ANC`). '>' means a dominance relation (from the head to the dependent), '<' an inverse relation. Between the '>' and '<' operators are the labels of this relation. If the dominance operators meet like this: '< >', then there is a common ancestor node (at this position in the path) that dominates both the center and the anchor (see the paths at ranks 5a and 9).

The most common path is `CNT > nsubj > ANC`, i.e., the anchor (a noun) functions as the subject of the center (a verb or predicative). An instance of this path is example (7), whose dependency annotation is shown in Figure 6.

Similar relations can be found on ranks 2, 5b, 7, 10a, where the anchor (a noun) is an argument or modifier of the verbal center. Or the anchor modifies a nominal center (rank 3).

The path `CNT > ANC` (rank 8) does not contain a function specification. These are cases of compound-internal relations as in example (9).

In the majority of these paths, the center dominates the anchor, confirming our hypothesis. An exception to our hypothesis is the path `CNT < amod < ANC` (rank 4), i.e. a metaphorical attributive adjective modifies an anchor noun – this configuration is frequently investigated in the literature (see Section 2). However, the reverse case, `CNT > amod > ANC`, also occurs (rank 10b), as also observed by Sullivan (2009) (see her examples in (4)).

[10] Note that some of the metaphor chains have no anchor because they are continued metaphors with the label `center+ctd`. Some others have no local anchor because an anchor was marked in an adjacent clause or sentence. In both cases no dependency paths could be extracted.

	Path	Freq	Perc
Metaphors			
1	CNT > nsubj > ANC	114	23.7
2	CNT > obj > ANC	73	15.2
3	CNT > nmod > ANC	28	5.8
4	CNT < amod < ANC	23	4.8
5a	CNT < obj < > nsubj > ANC	19	4.0
5b	CNT > obl:loc > ANC	19	4.0
7	CNT > obl:mod > ANC	17	3.5
8	CNT > ANC	12	2.5
9	CNT < obl:dir < > nsubj > ANC	9	1.9
10a	CNT > obl:arg > ANC	8	1.7
10b	CNT > amod > ANC	8	1.7
Comparisons			
1a	CNT < obl:mod < > nsubj > ANC	2	7.7
1b	CNT < obl:mod < > obj > ANC	2	7.7
1c	CNT < obj < > nsubj > ANC	2	7.7

Table 4: Top: The 10 most frequent dependency paths between a center (CNT) and its anchor (ANC) in metaphors, ranked according to frequency. Bottom: For comparisons, the 3 top frequent path are displayed; all other paths occur only once.

Path length	Mean	SD	Min	Max
Metaphors	1.3	0.6	0	4
Comparisons	2.3	1.0	1	5

Table 5: Average length of the dependency paths in all documents, measured as the number of edges between the center and the anchor. A length of 0 indicates compounds.

Other exceptions are the paths on ranks 5a and 9, in which the center and the anchor have a common ancestor.

In the comparisons, there are only three paths that occur more than once. In general, the paths in the comparisons are heterogeneous and tend to be longer than in the metaphors, see Table 5.

Overall, the paths in which the center dominates the anchor clearly predominate, see Table 6: in 2/3 of the metaphors, the center is the syntactic head of the anchor. In the control group, the comparisons, the largest group is the mixed group.

Metaphor type	C>A	A>C	Mixed
Metaphors (%)	66.5	10.4	23.1
Comparisons (%)	19.2	3.8	77.0

Table 6: Proportions of the different path types in metaphors and comparisons. 'C>A': center dominates anchor; 'A>C': anchor dominates center; 'Mixed': center and anchor have common ancestors.

5 Conclusion

The annotations in the sermons have confirmed the hypothesis that centers are typically the syntactic head of anchors. Counterexamples concern attributive adjectives, which, however, occur in both constellations. The result for the adjectives corresponds to the study by Sullivan (2009). A semantic analysis according to Construction Grammar would show whether the lemmas involved are instances of dependent vs. autonomous elements, as illustrated in example (4). In contrast to the study by Sullivan (2009), however, many of the (syntactic) dependency paths between source and target expressions are considerably more complex than the rather simple CxG construction types. This indicates that the dependency relations cannot be easily mapped to the constructions investigated by Sullivan (2009). It must therefore remain an open question whether an explanation in terms of CxG based on semantic properties (dependent vs. autonomous) could explain the syntactic properties.

A possible purely syntactic explanation could be that a syntactic head is more prominent than a dependent and, therefore, a (deliberate) metaphorical expression, which possibly requires more processing efforts than literal expressions, can be processed more easily in such a salient position.

Irrespective of this, the observations from this study may be useful for the automatic recognition of metaphors, by restricting the search space.

Acknowledgments

I would especially like to thank my co-annotators Alexandra Wiemann and Anna Maria Schröter for their annotations and the joint discussions. I am also very grateful to the anonymous reviewers for their helpful comments. Funded by/gefördert durch die Deutsche Forschungsgemeinschaft (DFG) – SFB 1475 – Projektnummer 441126958.

References

Yuri Bizzoni, Stergios Chatzikyriakidis, and Mehdi Ghanimifard. 2017. "Deep" learning: Detecting metaphoricity in adjective-noun pairs. In *Proceedings of the Workshop on Stylistic Variation*, pages 43–52, Copenhagen, Denmark. Association for Computational Linguistics.

William Croft. 1993. The role of domains in the interpretation of metaphors and metonymies. *Cognitive Linguistics*, 4(4):335–370.

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

Stefanie Dipper, Cora Haiber, Anna Maria Schröter, Alexandra Wiemann, and Maike Brinkschulte. 2024a. Universal Dependencies: Extensions for modern and historical German. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17101–17111, Torino, Italia.

Stefanie Dipper and Adam Roussel. 2024. Korpus deutschsprachiger evangelischer Predigten der Gegenwart, Version 0.2. `https://gitlab.rub.de/comphist/predigtenkorpus`.

Stefanie Dipper, Adam Roussel, Alexandra Wiemann, Won Kim, and Tra-my Nguyen. 2024b. Guidelines for the annotation of deliberate linguistic metaphor. In *Proceedings of the 4th Workshop on Figurative Language Processing (FigLang 2024)*, pages 53–58, NAACL, Mexico City, Mexico (Hybrid). Association for Computational Linguistics.

Lisa Gandy, Nadji Allan, Mark Atallah, Ophir Frieder, Newton Howard, Sergey Kanareykin, Moshe Koppel, Mark Last, Yair Neuman, and Shlomo Argamon. 2013. Automatic identification of conceptual metaphors with limited knowledge. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 328–334.

Cora Haiber. 2024. A crosslingual approach to dependency parsing for Middle High German. In *Proceedings of the 20th Conference on Natural Language Processing (KONVENS 2024)*, pages 23–31, Vienna, Austria. Association for Computational Linguistics.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics.

Saisuresh Krishnakumaran and Xiaojin Zhu. 2007. Hunting elusive metaphors using lexical resources. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*, pages 13–20, Rochester, New York. Association for Computational Linguistics.

George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. The University of Chicago Press, Chicago, London.

Yann Mathet, Antoine Widlöcher, and Jean-Philippe Métivier. 2015. The unified and holistic method gamma (γ) for inter-annotator agreement measure and alignment. *Computational Linguistics*, 41(3):437–479.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Ekaterina Shutova, Simone Teufel, and Anna Korhonen. 2013. Statistical metaphor processing. *Computational Linguistics*, 39(2):301–353.

Gerard J. Steen. 2008. The paradox of metaphor: Why we need a three-dimensional model of metaphor. *Metaphor & Symbol*, 23(4):213–241.

Gerard J. Steen, Aletta G. Dorst, J. Berenike Herrmann, Anna A. Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A Method for Linguistic Metaphor Identification. From MIP to MIPVU*. Number 14 in Converging Evidence in Language and Communication Research. John Benjamins, Amsterdam.

Karen Sullivan. 2009. Grammatical constructions in metaphoric language. In Barbara Lewandowska-Tomaszczyk and Katarzyna Dziwirek, editors, *Studies in Cognitive Corpus Linguistics*. Peter Lang.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 248–258, Baltimore, Maryland. Association for Computational Linguistics.

Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690, Edinburgh, Scotland, UK. Association for Computational Linguistics.

A New Hebrew Universal Dependency Treebank:
The First Treebank of Post-Rabbinic Historical Hebrew

Rachel Tal,[*,1,2] Shlomit Fuchs,[*,1] Orly Albeck,[2] Elisheva Brauner,[1,2]
Yitzchak Lindenbaum,[1,2] Ephraim Meiri,[1,2] Avi Shmidman[1,2]
[*]Equal contribution
[1]Bar-Ilan University, Ramat Gan, Israel
[2]DICTA, Jerusalem, Israel
rachel.tal@mail.huji.ac.il avi.shmidman@biu.ac.il
{shumital, orlyalbeck, efbrauner, yitzilindenbaum, ephraimmeiri}@gmail.com

Abstract

The corpus of post-Rabbinic historical Hebrew is a foundational corpus of Jewish heritage, containing over a billion words of legal, hermeneutical, and philosophic texts (and more). However, because the linguistic norms of the corpus diverge so often from that of modern Hebrew, the corpus cannot be computationally analyzed with existing Hebrew parsers. In order to fill this lacuna, we present the first Universal Dependencies corpus of post-Rabbinic historical Hebrew. The corpus comprises over 11,800 words, and we are pleased to release it to the community.

1 Introduction

The post-Rabbinic historical Hebrew corpus is comprised of over a billion words, authored across over a thousand years between the tenth and nineteenth centuries,[1] principally in European, Asian, and North African lands. It includes, inter alia, works of legal argumentation and responsa, commentaries on Scripture and Talmud, philosophical treatises, and even works on scientific matters.

The language employed in this corpus contains many unique linguistic characteristics which differentiate it from other layers of Hebrew, and thus pose a great challenge for computational analysis. Heretofore no syntax-annotated corpus has existed for this layer of Hebrew, and existing parsers for modern Hebrew fall flat when applied to this corpus.

In order to fill this lacuna and pave the way for computational analysis of this sizeable corpus which comprises the foundation of Jewish culture and law, we have embarked upon a new project – the first of its kind – to annotate a representative set of post-Rabbinic historical Hebrew sentences as per the Universal Dependencies standard. We are pleased to announce the completion of the inaugural batch of this corpus, and to release the corpus to the community.

2 Existing Hebrew Corpora

The Hebrew language is currently represented in four UD corpora. The first of these corpora, UD HTB (Sade et al., 2018), building on the work of the original HTB (Winter et al., 2001), provides 6143 sentences (114K tokens) of modern Hebrew, taken from the newspaper *Ha'aretz*. This corpus laid the groundwork for application of the UD guidelines to Hebrew, with regard to dependency relations and segmentation of space-delimited tokens into syntactic words. Zeldes et al. (2022) recognized the need for a more diverse corpus and created a new corpus ("IAHLTwiki") of 5K sentences (140K tokens) from 39 Hebrew Wikipedia articles spanning 7 domains. They suggest adjustments to the conventions used in UD HTB's conventions for both segmentation and dependency relations.[2] This was followed by the IAHLTKnesset treebank (2800 sentences, 67K tokens), drawn from protocols held in the Israeli parliament between 1998 and 2022, further improving the diversity of the available corpora with the addition of spoken language (Goldin et al., 2024). All three of these treebanks cover only the most recent half-century of Hebrew. Swanson and Tyers (2022) began to rectify this with a corpus of Biblical Hebrew, consisting of 2.5K sentences from the books of Genesis, Exodus, Leviticus, and Ruth, totaling 62K tokens. The intervening two millennia of Hebrew, then, remained unspoken for.

[1] To be sure, there is a wide range of linguistic styles within this corpus, and this corpus can certainly be subdivided into multiple subdivisions (Goshen-Gottstein, 1985; Tènè, 1985). Nevertheless, on the whole, scholars have differentiated between four primary layers of Hebrew: Biblical Hebrew, Rabbinic Hebrew, post-Rabbinic historical Hebrew, and modern Hebrew, and it is this division which motivates our paper (Ben-Hayyim, 1985; Rabin, 1985).

[2] For maximum compatibility, we release our new corpus both in HTB format as well as IAHLT format.

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 91–96
August 28-29, 2025 ©2025 Association for Computational Linguistics

3 The Present Corpus

The present corpus comprises over 11,000 words of text from a variety of post-Rabbinic Hebrew sources. Full details of the sources and sentence/word counts are provided in Table 1. Each sentence was initially annotated in terms of its syntactic functions and dependencies by one of our four linguistic experts (the first four authors of the present paper). Afterward, the four linguists convened together and critically reviewed each annotation, adjusting and honing the annotations to ensure both accuracy and consistency.[3]

4 Unique Syntactic Characteristics of Post-Rabbinic Historical Hebrew

As noted, from a linguistic perspective, the norms of post-Rabbinic historical Hebrew are quite different from that of modern Hebrew; hence the need for a new annotated corpus. In this section we survey three such differences.

4.1 Dislocated elements

The dislocated elements in our corpus are primarily cases of topicalization. In post-Rabbinic texts, dislocated topicalization is very common, much more so than in modern Hebrew. Thus, in IAHLTwiki there are 24 dislocated elements in the entire corpus, and only four of them are cases of topicalization (0.04% of the sentences), and in HTB there are 13 cases of dislocated topicalization (0.21%). In IAHLTknesset, which contains many transcriptions of spoken Hebrew, the frequency of dislocated topicalizations is a bit higher, yet still limited to one percent (29 sentences out of 2883). By contrast, in our corpus, there are 37 dislocated elements (11.5%) almost all of which are cases of topicalization.

To the limited extent that dislocated elements do occur in modern Hebrew in written form, they generally appear with a comma (or other punctuation mark) which indicates the boundary of the extraposed part. In post-Rabbinic historical Hebrew, there are generally no punctuation marks. Therefore, a syntax parser trained on the existing modern Hebrew corpora is likely to fail to locate the extraposed part and identify it as a dislocated element. Consider the following sentence:[4]

(1) Ḳaṭan ha-yodeaʿ lehitʿaṭef
small DET/SCONJ-knows to.wrap
aviṿ tsarikh liḳaḥ lo tsitsit
his.father must to.take to.him tsitsit
leḥankho
to.educate.him
‘A small boy who knows how to wrap himself [in a tallit] his father must buy him [a tallit with] tsitsit to educate him.’
[Sentence ID: 241]

The dislocated element, "a small boy who knows how to wrap himself," appears, due to the lack of punctuation, as though it were the subject of the sentence. In fact, however, the subject is "his father", meaning the father of that boy. This structure appears here, as in many other instances in our corpus, in place of a conditional clause ("If a boy knows how to wrap himself, his father must take. . . ").

Indeed, when running this sentence through the DictaBERT syntactic parser (Shmidman et al., 2024) – a syntax parser trained on modern Hebrew – it fell into this very trap. It labeled both *ḳatan* (a small boy) and *av* (father) as `nsubj`, each dependent on words within the main component of the sentence (*tsarikh* and *liḳaḥ*, respectively). This results in an illogical dependency parsing.

4.2 Causal clauses introduced by "*she-*" alone

Causal clauses in Hebrew begin with various connective words, including *ki*, *mi-pene she-*, *mishum she-*, and others. In rare cases in biblical language and more commonly in post-Rabbinic Hebrew, we find the prefix *she-* and its (originally Aramaic) equivalent *de-* used as a subordinating conjunction for causal clauses without a preceding connective word. The appearance of *she-*/*de-* without a connective word creates a structure which normally indicates a relative clause, and thus, in most such instances, the correct analysis of the sentence can only be determined based on semantics.

Such causal clauses appear in our corpus more than 15 times (more than 5% of sentences). By contrast, causal clauses with this syntax are highly irregular in modern Hebrew, and do not exist in the available annotated corpora of modern Hebrew.[5]

[3] Because the corpus comprises less than 20K words, we have not performed a train-dev-test split, as per `https://universaldependencies.org/release_checklist.html`; rather, we recommend testing via 10-fold cross validation.

[4] Example sentences in the linguistic discussions in this paper are presented in a format that maximizes readability, but which sometimes diverges from the UD segmentation (unless otherwise indicated). For the proper UD tokenizations and tagging of the example sentences, please reference the corpus using the provided sentence IDs.

[5] To be sure, other (non-causal) types of adverbial clauses

Work	Sentences	Words	Author	Author d.	Location	Description
Igeret orhot olam	13	1084	Abraham Farissol	1525	Italy	Geographic/cosmographic studies.
Sefer ha-hinukh	14	675	Unknown	~13th C.	Spain	Treatise on Biblical Commandments.
Rashi la-Torah	100	2161	Shlomo Yitzchaki	1105	France	Rashi's Pentateuchal Commentary.
Shulhan arukh	80	4015	Joseph Karo	1575	Israel	15th century code of Jewish Law.
Sefer Maharil	18	569	Yaakov HaLevi Moelin	1427	Germany	Record of Ashkenazic Customs
Miscellaneous	92	3288	—	—	—	Eclectic collection of sentences from throughout post-rabbinic literature.
TOTAL	318	11802	—	—	—	—

Table 1: A summary of the texts included within our annotated post-Rabbinic historical Hebrew UD corpus.

Therefore, a syntactic analyzer of modern Hebrew has difficulty dealing with it. For example[6] (2):

(2) ṿe-en omrim kol yeme nisan tsidḳatkha
and-no saying all days.of Nisan Tsidkatkha
de-dino kemo teḥinah
SCONJ-its.status like supplication
'And "Tsidkatkha" ("Your righteousness") is not recited [during] all the days of the month of Nisan, for its status is that of a supplication.'
[Sentence ID: 131]

We demonstrate the difficulty by running the sentence through the DictaBERT parser (Shmidman et al., 2024) (see Figure 1). In Figure 2 we present the correct analysis, as we have analyzed it in our corpus.

4.3 Conjunctions

In our corpus, we often find verbal elements within a single clause which are not of the same tense, yet are joined by coordinating conjunctions, as in (3), (4). Such a conjunction is expected between different sentences, but not within the same sentence. In modern Hebrew, such a conjunction is very rare, if not non-existent. We conducted extensive searches in the existing Hebrew treebanks and found no such conjunctions. For example:

(3) ha-holekh ba-derekh ṿe-higiaʿ
DET/SCONJ-walking in.the-way and-arrived
la-ʿir ṿe-rotseh lalun bah
to.the-city and-wishes to.lodge in.it
'[One] who travels and arrived at a city and wishes to lodge therein.'
[Sentence ID: 295]

The sentence begins with a present participle, "ha-holekh" (Adler et al. 2008), and continues with a past tense verb "ve-higiaʿ".[7] These two words together, in coordination, comprise the root of the syntactic subject of the clause, and we would expect them to be of the same tense. Alternatively, the conjunction could have been replaced by a relative pronoun. The use of coordination here diverges from normative syntax of modern Hebrew.

(4) tsarikh leha'arikh be-ḥet shel eḥad ...
must to.lengthen in-Ḥ of EḤAD ...
ṿe-ya'arikh be-dalet shel eḥad
and-he.will.lengthen in-D of EḤAD
'One must hold (i.e. tenuto) the h of "ehad" (=one) . . . and will hold the d of "ehad".'
[Sentence ID: 288]

The legal imperative in Hebrew can be expressed in several ways, e.g. by impersonal verb (see: Mor and Pat-El 2016) or future tense. We would expect to find a single mode in a given citation, but here we have a mixture of the two – impersonal verb ("tsarikh le-ha'arikh") and future tense ("ṿe-ya'arikh").

5 Annotation Decisions

5.1 UD tags specific to this corpus

In order to capture the linguistic complexities of this corpus, we have added a number of new features to the UD annotation. All the new features are materialized as subtypes of existing UD tags; thus the corpus remains valid for crosslingual comparison, as illustrated in Swanson et al. (2024).

5.1.1 part

The use of the Hebrew participle effects unique syntactic constructions. This is because, on the

are occasionally subordinated by a *she-* alone, and these do appear in those corpora, albeit very rarely. Regarding *she-*clauses in general, see Kogut (1937). *De-* does not appear in modern Hebrew.

[6] We have brought here a sentence that actually uses the Aramaic *de-* in the original text; however, because this sentence will be used to show the inability of the modern Hebrew parser to analyze such sentences, we adjusted to the Hebrew *she-* in order to give the parser a fighting chance.

[7] Regarding the double gloss of the Hebrew "ha" clitic at the start of the sentence: this clitic straddles the boundary between a definite article and a relativizer. In practice, in the corpus, any given instance of the clitic is specified as either SCONJ or DET, because only a single value can be selected.

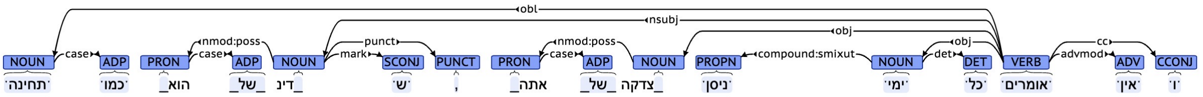


Figure 1: The sentence as analyzed by a syntactic parser trained on modern Hebrew (in right-to-left reading order). The causal clause is analyzed by the parser as two separate parts: its subject is parsed as the subject of the main sentence; its predicate is parsed as an oblique argument of the main verb; the subordinator '*she*-' is illogically tagged as a mark to the subject of the subordinate clause.

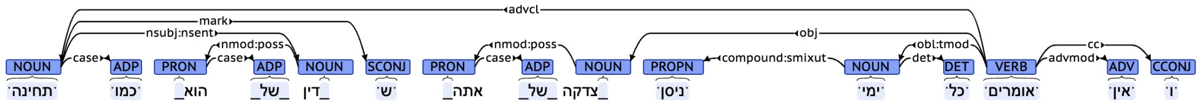


Figure 2: The sentence as annotated in our corpus by our linguistic experts.

one hand, it is a verbal form, but on the other hand, it can serve as a nominal or adjectival component. (Rosén 1956; Zewi and Reshef 2009; Sharvit 1980; Adler et al. 2008).

Consider the participle at the beginning of (3). The part-of-speech of "holekh" ("is walking"), as a participle, is VERB. In a sense, it is the root of a clause that complements an assumed nominal "one": "[one] who is walking in the way". Yet, the word in question is the subject of the sentence, and therefore tagged as nsubj.[8] Furthermore, the "ha" clitic attached to it normally serves as a definite article attaching to nominals.[9] That is, the obl "way" is, remarkably, a complement of a word with notable nominal morphosyntactic characteristics. Moreover, participles appear in our corpus in nominal positions other than nsubj; for instance, as the second member of a genitive construct (*smikhut* – a Hebrew construct that connects two nominals).[10]

In all of the aforementioned cases, the participle straddles the border between verb and noun, and an annotation that ignores this would be misleading and inconsistent. In order to bridge this gap and to properly represent the complexity of these participle forms, we have labeled their own dependency relations according to their nominal syntactic function, while adding the part subtype to their dependents that function as verbal complements and thus reflect the verbal character of the participle.

5.1.2 conj:push

Many times in post-Rabbinic Hebrew we find a series of coordinating conjunctions which are not equal in their syntactic value. For instance, coordinating conjunctions often appear where we would expect a subordinating conjunction, such that we end up with a case of nested coordination (Universal Dependencies). In order to capture this nuance, we have added a push subtype; specifically, in a structure of type (A, B), C, the push subtype is specified for B.

5.2 Tokenization of negative particles

In negative particles of type "*en* ('no') + personal pronoun", the pronominal component sometimes serves as the subject of the sentence (see e.g. (5)). In other cases, however, it simply negates a sentence that has an explicit subject (with which the pronominal component agrees – see e.g. (6)). In the former case, the particle contains two syntactic words; in the latter, it contains one. In existing modern Hebrew treebanks, (see Section 2), such particles are always tagged as a single word. In the biblical Hebrew treebank (ibid.), they are always split.[11] We have chosen to differentiate between the cases: when the sentence contains no other explicit subject, we split the token into the negating particle *en*, which receives an advmod dependency relation, and the corresponding pronoun, which receives an nsubj dependency relation. When the

[8] Here we follow the UD guidelines for Hebrew (Universal Dependencies Contributors, 2024), under which such participles are tagged nsubj and not csubj.

[9] See footnote 7.

[10] See sentence 173 in the tagged corpus for such an example. See also sentence 127, in which a participle has both obl and case dependents.

[11] This seems to follow the approach that pronouns in biblical Hebrew do not normally serve as copulae (for discussion see Joüon and Muraoka 2011, §102*k* and §154*i*). In this approach, all sentences with an explicit subject and a negative particle with a pronominal suffix are naturally analyzed as *casus pendens* – that is, what appears to be the explicit subject is in fact a dislocated element, and the negative particle contains the subject.

subject appears, we segment as in the modern Hebrew treebanks. An example of our segmentation (with selected morphological features added for clarity) for each kind of sentence appears below.

(5) en-i rotseh lehinaḳem mi-Ḳayin ʿakhshaṿ
no-I want to.be.avenged from-Cain now
‘I do not want to take revenge on Cain now.’
[Sentence ID: 153]

(6) ḳṭanah **enah** yekholah laʿaśot shaliaḥ
small.FSG **no.3MSG** able to.make messenger
‘A minor girl cannot appoint a proxy.’
[Sentence ID: 313]

6 Conclusion

We have prepared the first UD-tagged corpus of historical post-Rabbinic Hebrew, containing over 11,000 words across multiple genres and time periods. We are pleased to release this corpus to the public. The corpus is released under a Creative Commons Attribution 4.0 International License (CC BY 4.0). The corpus is available now on github,[12] and we hereby submit the corpus to the upcoming UD release as a new treebank within the *heb* language.

Acknowledgements

This work has been funded by the European Union (ERC, MiDRASH, Project No. 101071829; Principal investigators: Nachum Dershowitz, Tel-Aviv University; Judith Olszowy-Schlanger, EPHE-PSL; Avi Shmidman, Bar-Ilan University, and Daniel Stoekl Ben Ezra, EPHE-PSL), for which we are grateful. Views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

Meni Adler, Yael Dahan Netzer, Yoav Goldberg, David Gabay, and Michael Elhadad. 2008. Tagging a Hebrew corpus: the case of participles. In *LREC*.

Zeev Ben-Hayyim. 1985. The problems of the unity of the Hebrew language throughout its history and its periodization. In Moshe Bar-Asher, editor, *Language Studies*, volume 1. Magnes.

Gili Goldin, Nick Howell, Noam Ordan, Ella Rabinovich, and Shuly Wintner. 2024. The Knesset corpus: An annotated corpus of Hebrew parliamentary proceedings.

Moshe Goshen-Gottstein. 1985. Corpus, genre and the unity of Hebrew – aspects of conceptualization and methodology. In Moshe Bar-Asher, editor, *Language Studies*, volume 1. Magnes.

Paul Joüon and Takamitsu Muraoka. 2011. *A Grammar of Biblical Hebrew*. Gregorian and Biblical Press, Roma.

Simcha Kogut. 1937. *The Complex Sentence in Sefer Hasidim*. Ph.d. dissertation, Hebrew University of Jerusalem. See especially pp. 204–207.

Uri Mor and Na’ama Pat-El. 2016. The development of predicates with prepositional subjects in Hebrew. *Journal of Semitic Studies*, 61(2).

Chaim Rabin. 1985. Periods of the Hebrew language. In Moshe Bar-Asher, editor, *Language Studies*, volume 1. Magnes.

Haiim B. Rosén. 1956. *Our Hebrew: Its Character in Light of Linguistic Methods*. Am Oved, Tel Aviv.

Shoval Sade, Amit Seker, and Reut Tsarfaty. 2018. The Hebrew Universal Dependency treebank: Past present and future. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Shimon Sharvit. 1980. The tense system in Mishnaic Hebrew. In G. Sarfatti, P. Artzi, H. Y. Greenfield, and M. Z. Kaddari, editors, *Studies in Hebrew and Semitic Languages Dedicated to the Memory of Prof. Yechezkel Kutscher*, pages 110–125. Bar-Ilan University, Ramat Gan.

Shaltiel Shmidman, Avi Shmidman, Moshe Koppel, and Reut Tsarfaty. 2024. MRL parsing without tears: The case of Hebrew. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4537–4550, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Daniel Swanson and Francis Tyers. 2022. A Universal Dependencies treebank of Ancient Hebrew. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2353–2361, Marseille, France. European Language Resources Association.

Daniel G Swanson, Bryce D Bussert, and Francis Tyers. 2024. Producing a parallel universal dependencies treebank of Ancient Hebrew and Ancient Greek via cross-lingual projection. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13074–13078.

[12] https://github.com/ERC-Midrash/UD_Hebrew-PostRab

David Tènè. 1985. Historical identity and unity of Hebrew and the division of its history into periods. In Moshe Bar-Asher, editor, *Language Studies*, volume 1. Magnes.

Universal Dependencies. Universal dependencies - nested coordination. Accessed: 2025-04-07.

Universal Dependencies Contributors. 2024. UD for Hebrew. Accessed: 2025-06-12.

Yoad Winter, Alon Altman, K. Sima'an, Alon Itai, and Noam Nativ. 2001. Building a tree-bank of modern hebrew text.

Amir Zeldes, Nick Howell, Noam Ordan, and Yifat Ben Moshe. 2022. A second wave of UD Hebrew treebanking and cross-domain parsing.

Tamar Zewi and Yael Reshef. 2009. The active participle and temporal expression in Hebrew. *Leshonenu*, 71(3–4):315–344.

Syntax of referents of relative markers: Evidence from a corpus of learner English

Izabela Czerniak
Åbo Akademi University
Tehtaankatu 2
20500 Turku, Finland
izabela.czerniak@abo.fi

Debopam Das
Åbo Akademi University
Tehtaankatu 2
20500 Turku, Finland
debopam.das@abo.fi

Abstract

We investigate the referents of relative markers of English relative clauses, focusing on their syntactic role in the matrix clauses. The referents, unlike relative markers and related features, have compratively remained understudied. We examine the syntactic environments of the referents as part of a larger project, which develops the ICLE-RC, a corpus of learner English texts annotated for relative clauses and related phenomena (it-/pseudo-clefts, existential-relatives, etc.). The corpus derives from the International Corpus of Learner English (ICLE; Granger et al., 2020), and contains 144 academic essays, representing six L1 backgrounds – Finnish, Italian, Polish, Swedish, Turkish, and Urdu. We annotate those texts for over 900 relative clauses (and over 400 related phenomena), with respect to a wide array of lexical, syntactic, semantic, and discourse features. Results from our analysis show that the relativisation of referents varies according to their syntactic functions. The referents are also observed to interact with other RC-features, yielding systematic variations across different L1 backgrounds, (some of) which can potentially be attributed to the typological properties of the associated L1.

1 Introduction

Relative clauses (henceforth RCs) are a type of subordinate clauses that typically modify nouns or noun phrases, and sometimes also adjectives[1], adverbs[2], PPs[3], VPs[4], and even entire clauses[5]. RCs constitute a rich body of research, addressing themes such as syntactic and typological variation (Comrie, 1998; Grosu, 2012), semantic features (Cornish, 2018), discourse functions (Brandt et al., 2009), FLA/SLA (Diessel and Tomasello, 2005; Doughty, 1991), parsing (Goad et al., 2021), processing (Reali and Christiansen, 2007), historical usage (Suárez-Gómez, 2006), diachronic development (Leech et al., 2009; Fajri and Okwar, 2020), corpus-based analysis (Biber et al., 1999; Weichmann, 2015), and World Englishes (Suárez-Gómez, 2015). Despite the depth and breadth of previous research, the scope of these studies have largely remained confined to the analysis of RCs alone and associated features found therein.

We strive to extend the scope of RC analysis, by examining the larger syntactic environment in which RCs occur. In particular, we investigate the referents of relative markers of English RCs, focusing on their syntactic role in their respective matrix clauses. We examine RC-referents as part of a larger project, the ICLE-RC, which builds a corpus of learner English annotated for RCs and related phenomena (it-/pseudo-clefts, existential-relatives, etc.). The corpus builds on a subset of the International Corpus of Learner English (ICLE; Granger et al., 2020), and contains 144 academic essays, representing six L1 backgrounds – Finnish, Italian, Polish, Swedish, Turkish, and Urdu. In this paper, we present our multi-layered, feature-rich annotation framework for RC(-referent) analysis, and report on our corpus analysis of RC-referents and their interaction with other RC-features.

This paper is structured as follows: We outline the previous work on RC-referents in Section 2. Section 3 introduces our large-scale corpus project, and describes the annotation schemes. We present the general results and those for referent functions in Section 4 and Section 5, respectively. Section 6 discusses the findings, and Section 7 concludes the paper, outlining some future research directions.

2 Previous work

One of the most influential work on RCs (and RC-referents) is offered by Keenan and Comrie's

[1] *Pat is [beautiful], which, however, many consider her not.*
[2] *He moved [abroad] where he found a good job.*
[3] *He found a body [under the bridge] where nothing grows.*
[4] *She told me to [design it myself], which I simply can't.*
[5] *[Alex bought a mansion], which made him bankrupt.*

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 97–110
August 28-29, 2025 ©2025 Association for Computational Linguistics

(1977) NP accessibility hierarchy (NPAH):

(1) subject > direct object > indirect object > oblique > genitive > comparative

NPAH stipulates that languages that relativise on one position on the hierarchy will also relativise on the positions above it. According to this scale, the subjects of the matrix clause are most prone to be relativised, followed by the direct objects, which is then to be followed by the indirect objects, and so on. The validity of NPAH is supported by numerous studies on RCs across languages, rendering it one of the few putative typological universals[6].

Besides NPAH, there exist some studies that considered RC-referents an important RC-feature. For example, Fox and Thompson (1990) investigated the syntactic and discourse properties of the head NPs in the matrix clause and their interaction with RCs in conversations. They observed that the structuring of RCs is crucially shaped by the formulation of the referents according to many interactive and cognitive factors of the communicative situations.

Tagliamonte et al. (2005) examined relative markers in vernacular varieties of British English, and observed the prevalence of *that* and zero marker, instead of *wh*-forms. The authors identified the type of the referent (e.g., definite or indefinite NPs) as one of the determining factors behind the marker preference. More particularly, indefinite referents (along with sentence structure) entailed the use of the zero-variant in RCs.

Hinrichs et al. (2015) investigated the changing trends in the use of restrictive relativisers, examining the shift from *which* to *that* in written standard English. The authors conducted a multivariate analysis on a large collection of RCs (16K+) from the Brown corpus[7], and used a number of independent variables which included, among other features, a set of referent-features, such as the POS, number, length, and definiteness of the referent. The study concluded that the shift (*which* → *that*) took place largely under the influence of American English and was regulated by various prescriptivism-related factors.

3 The ICLE-RC project

We have developed the ICLE-RC to investigate RCs and related phenomena (it-/pseudo-clefts, existential-relatives, etc.) in learner English. The corpus builds on a subset of the International Corpus of Learner English (ICLE; Granger et al., 2020). The ICLE is a corpus of academic essays written by undergraduate students from a set list of topics[8]. These students are intermediate or advanced learners of English, coming from different L1 backgrounds. The first version of the ICLE-RC contains 144 ICLE texts (100K+ words), covering six L1 backgrounds – Finnish, Italian, Polish, Swedish, Turkish, and Urdu – with 24 texts from each[9]. These texts are annotated for 924 RCs, with respect to a wide array of lexical, syntactic, semantic, and discourse features. These texts are also annotated for 407 related phenomena, which we call *other constructions* (henceforth OCs)[10].

The ICLE-RC is designed to serve a number of purposes. First, the corpus provides real language data to assess English learners' use of RCs against the standard rules of English grammars (e.g., the use of *which* for a human referent, or the use of a comma for integrated RCs). Second, the ICLE-RC covers six L1 backgrounds representing six different language families (Pereltsvaig, 2023) – Finnish: Uralic; Italian: Romance; Polish: Slavic; Swedish: Germanic; Turkish: Turkic; and Urdu: Indo-Aryan[11]. This would allow identifying typological patterns for certain RC features as well as highlighting those which potentially result from cross-linguistic influence (e.g., the use of extraposed RCs). This would also offer significant implications for research in World Englishes, in comparison to native varieties of English (e.g., by comparing the ICLE-RC with comparable corpora such as ICNALE (Ishikawa, 2023) as well

[6] Nevertheless, counter-evidence to NPAH (e.g., the uniformity of the subject-object asymmetry) has been provided by some later studies. For an overview, see Kidd (2011).

[7] `https://varieng.helsinki.fi/CoRD/corpora/BROWN/`

[8] Some of the ICLE essay topics are as follows: (1) *The prison system is outdated.*, (2) *No civilised society should punish its criminals: it should rehabilitate them.*, (3) *Feminists have done more harm to the cause of women than good.* For specimen essays, check out the ICLE500 dataset.

[9] The detailed distribution of the essays in the ICLE-RC is provided in Table 11 in the Appendix.

[10] OCs either resemble RCs (particularly because of the use of words such as *that*, *which*, or *who*) but are not RCs proper, or they are a special type of RCs. OCs comprise four major types, as follows:

it-cleft: *It was only last year that he got his tenure.*
pseudo-cleft: *What I need is a long cool drink.*
relative-there: *There was one man that kept interrupting.*
fused-relatives: *The dog ate what I had left on my plate.*

[11] The selection yields four Indo-European and two non-Indo-European languages.

as those of native academic English such as LOCNESS (Granger, 1998)). Finally, the corpus would help us explore English learners' use of OCs as alternative strategies of information structuring, in addition to RCs.

3.1 Main annotation framework

In the ICLE-RC, we have annotated the RCs[12] for a wide range of lexical, syntactic, semantic, and discourse features, as listed in Table 1. The complete taxonomy of the annotation features is provided in Table 12 in the Appendix.

Here, we first outline the main annotation features, except the grammatical functions of the referent (REFERENT FUNCTION), which is described in greater detail in the next sub-section[13].

RELATIVE MARKER (RM): RMs include the subordinator *that* and *wh*-words (e.g., *which*, *who*, *whose*) that introduce an RC. An additional feature *zero* is recognised to mark the absence of an overt RM for bare-relatives. These categories are exemplified below[14].

(2) Our duty should be to select programmes and to see only things ***that** open our mind.* [Italian; ITRS-1002]

(3) Those, ***who** cannot afford advertising campaigns led on a large scale*, have no chances of achieving success in any kind of business. [Polish; POLU-1006]

(4) **The status** ø *English has acquired today* is so dominant that it seems unlikely that ... [Finnish; FIJO-1003]

MARKER FUNCTION: This feature identifies the grammatical function of the relativised item (represented by the RM) in the RC. It comprises nine categories, largely adapted from Huddleston and Pullum (2002): `subject`, `direct object`, `indirect object`, `predicative complement`, `genitive subject determiner`, `predicate`, `complement of auxiliary verb`, `head of a to-infinitival VP`, and `adjunct`. For illustration, we here define and exemplify only three of those types (for more information about all categories and sub-categories, see Table 12).

`subject`: The relativised item functions as the subject in the RC, as in (5).

(5) These teachers ***who** want to prevent cheating* were once students. [Turkish; TRCU-1004]

`genitive subject determiner`: The relativised item (*whose*) is the genitive determiner in the subject NP of the RC, as in (6).

(6) ... his proposal is not only urgent but necessary as well for a democracy ***whose** purpose consists of controlling any political power.* [Italian, ITRS-1004]

`adjunct`: The relativised item functions as an adjunct or part of an adjunct in the RC, as in (7).

(7) ... the newspapers have talked about childporno and the right to have in one's possession videos or photos ***where** children are being exploited.* [Finnish; FIJY-1006]

REFERENT TYPE: The referent can be an entity, an abstract entity, or a proposition (a full clause). Furthermore, an entity can either be human or nonhuman. Examples of human, non-human, and abstract entity are given in (3), (7), and (6), respectively. (8) illustrates the proposition category.

(8) ... the product not advertised does not exist for customers, ***which** means it brings no profits.* [Polish; POLU-1006]

RESTRICTIVENESS: This feature identifies whether an RC is integrated or supplementary[15]. An integrated RC is an integral part of the referent NP that contains it, as in (9). A supplementary RC, by contrast, is characterised by a weaker link to its referent or surrounding structures, as in (10).

(9) The people ***who** happened to fall victim to this shameful disease* were persecuted. [Polish; POLU-1007]

[12] We have only annotated full RCs, and exclude reduced RCs on grounds of parsing and processing difficulties (Acuña Fariña, 2000; McKoon and Ratcliff, 2003).

[13] We exclude from the description two main RC-features, EMBEDDING and EXTRAPOSITION, as they are not central to the RC-referent analysis (and also because of the space constraint). For the same reason, we also do not include the annotation framework for OCs. For the detailed annotation guidelines, visit the project website.

[14] **Conventions for examples:** The RC is in italics; the RM is in bold; the referent is underlined. In case of RM-zero, there is no overt RM, and the referent is marked in bold instead. The text inside the square brackets lists the L1 background and the file number of the source text. **Note:** Some examples contain grammatical/spelling errors (as written by L2 students).

[15] The integrated-supplementary division of RCs corresponds to the distinction between restrictive and nonrestrictive RCs (hence the feature name is 'restrictiveness'). For the differences between these two dichotomies, see Huddleston and Pullum (2002).

#	feature	examples (of sub-features)	feature type
1	relative marker (RM)	*that*, *which*, *who*, zero	lexical/syntactic
2	grammatical function of referent	subject, object, predicative complement	syntactic
3	grammatical function of RM	subject, object, adjunct	
4	embedding of RC	embedded, non-embedded	
5	extraposition of RC	extraposed, non-extraposed	
6	type of referent	human, abstract entity	semantic/discourse
7	restrictiveness	integrated, supplementary	syntactic/discourse

Table 1: Primary categories of RC annotation

(10) ... I haven't mentioned about inequality in the social life, ***which*** *is the extension of inequality in the family life*. [Turkish; TRCU-1003]

3.2 The referent function sub-scheme

The REFERENT FUNCTION feature identifies the grammatical function of the referent of the RM in the matrix clause. It includes seven broad categories and fifteen specific sub-categories, as shown in Table 2[16]. These sub-categories are described below.

category	sub-category
`subject`	`subj-head-n`
	`in-subj-comp`
	`in-sub-adjunct`
`direct object`	`dir-obj-head-n`
	`in-dir-obj-comp`
	`in-dir-obj-adjunct`
`indirect object`	`indir-obj-head-n`
	`in-indir-obj-comp`
	`in-indir-obj-adjunct`
`predicative complement`	`pred-comp-np`
	`pred-comp-adj`
	`pred-comp-pp`
`adjunct`	`adjunct`
	`in-adjunct`
`clause`	`clause`

Table 2: REFERENT FUNCTION sub-scheme

`subj-head-n`: The head noun of the subject NP of the matrix clause is the referent. (If there is any complement and/or adjunct within that NP, the whole NP is considered as the referent.)

(11) **The third type of advertisement** ø *I do not like* is concerned to the tobacco business. [Italian; ITBO-1001]

`in-subj-comp`: An NP which is part of a complement within the subject NP is the referent.

(12) A secret to a slim figure, ***which*** *is a dream of many*, surely does not lie in fast food. [Polish; POLU-1008]

`in-subj-adjunct`: (An NP which is part of) an adjunct within the subject NP is the referent.

(13) All the informations are [sic], even the minor ones ***that*** *are seen unimportant*, are the chains of each other. [Italian; TRME-3006]

`dir-obj-head-n`: The head noun of the direct object NP in the matrix clause is the referent.

(14) We must look into ourselves and forget all the boring scientific theories ***which*** *have taken hold of our sense of reality ...* [Swedish; SWUL-1005]

`in-dir-obj-comp`: An NP which is part of a complement in the direct object NP is the referent.

(15) The main objection is the fact that it creates the demand for things ***that*** *people do not need*. [Polish; POLU-1006]

`in-dir-obj-adjunct`: (An NP which is part of) an adjunct in the direct object NP is the referent.

(16) According to that great king ... people ... should be punished by imposing on them the penalty equal in quality **to the criminal offences** ø *those people were charged with*. [Polish; POSI-1001]

`indir-obj-head-n`: The head noun of the indirect object NP in the matrix clause is the referent.

[16] Each feature under `predicative complement` is divided into further sub-types. For the complete annotation scheme, see Table 12 in the Appendix.

(17) If only done properly, mining and timbering... bring lots of revenue to **the state** ø *they live in*. [Swedish; SWUL-1006]

`in-indir-obj-comp`: An NP which is part of a complement within the indirect object NP is the referent.

(18) Thomas Sternes Eliot published 'The Waste Land' in 1922 and owes its final shape to the collaboration of Ezra Pound ***who*** *actually corrected it* ... [Italian; ITRS-1030]

`in-indir-obj-adjunct`: (An NP which is part of) an adjunct within the indirect object NP is the referent.

(19) John sent his letter to the professor of history with 100 publications, *some of* ***which*** *are quite remarkable*. [our example][17]

`pred-comp-np`: The referent is (part of) an NP that serves as the predicative complement in the matrix clause.

(20) Unfortunately, life is not a situation comedy ***where*** *every problem is happily solved*. [Italian; ITTO-1002]

`pred-comp-adjp`: The referent is (part of) an AdjP that serves as the predicative complement in the matrix clause.

(21) The world is full of ambitious and resolute persons ***who*** *are at the some time reliable and sensitive*. [Polish; POLU-1003]

`pred-comp-pp`: The referent is (part of) an PP that serves as the predicative complement in the matrix clause.

(22) It is like a chain process *in* ***which*** *better cures are required* ... [Polish; POSI-1004]

`adjunct`: The referent is an adjunct phrase in the matrix clause.

(23) Nobody is happy in a dictatorship ***where*** *violence and hypocrisy reigns* [sic]. [Swedish; SWUV-3003]

`in-adjunct`: An NP that is part of an adjunct in the matrix clause is the referent.

(24) In a family, ***which*** *is made up by four people*, there are at least two cars. [Italian; ITBO-2001]

`clause`: The whole matrix clause is the referent.

(25) In some countries homosexual marriages have been recently legalised, ***which*** *of course gave rise to many protests*. [Polish; POLU-1007]

An example of the ICLE-RC annotation is provided in Table 13 in the Appendix.

4 General results

The purpose of developing the ICLE-RC is to offer gold-standard data, and hence, the corpus is entirely created from human annotation. The RCs and OCs in the ICLE-RC were annotated by two annotators (the authors), who have many years of experience with various kinds of linguistic annotation. The annotation was performed using the UAM CorpusTool (version 2.8.16) (O'Donnell, 2008), and is saved in a stand-off XML format.[18]

The reliability of the annotation was tested through an IAA study. The two annotators independently annotated all 24 texts for the Polish part of the corpus. Given our multi-layered, feature-rich annotation scheme (Table 12), we calculated agreement only for the seven broad RC features: RM, REFERENT FUNCTION, MARKER FUNCTION, EMBEDDING, EXTRAPOSITION, REFERENT TYPE, and RESTRICTIVENESS. According to Cohen's kappa (Landis and Koch, 1977), agreement was almost perfect for REFERENT FUNCTION and MARKER FUNCTION (0.86, 0.80), substantial for RM and REFERENT TYPE (0.77, 0.73), and moderate for RESTRICTIVENESS (0.58)[19]. For the remaining two features, EMBEDDING and EXTRAPOSITION, prevalence prevented the calculation of meaningful κ-values. The agreement score was 89.35% for both features.[20]

The essays from different L1 backgrounds in the ICLE-RC vary with respect to the number of

[17] No token for this category was found in our corpus.

[18] For more information about the prospects of pre-annotating the ICLE-RC for syntactic (dependency) parses and the feasibility of (semi-)automating the RC annotation, see Das et al. (to appear).

[19] Previous research (Bache and Jakobsen, 1980; Hundt et al., 2012) also addressed the challenge of determining restrictiveness.

[20] For a detailed discussion about the reliability of the ICLE-RC annotation, see Das et al. (to appear).

words and sentences, as shown in Table 3. For example, on average the students with Finnish L1 produced the lengthiest essays (867.04 words per essay) while the students with Swedish L1 produced the shortest essays (664.29 words per essay)[21], although both groups produced sentences of almost equal length (about 22 words per sentence).

L1	# avg words	# avg sentences	# avg words per sentence
Finnish	867.04	39.38	22.02
Italian	718.33	27.21	26.40
Polish	705.92	33.17	21.28
Swedish	664.29	29.34	22.61
Turkish	786.75	39.25	20.04
Urdu	711.29	43.29	16.43
AVG	742.27	35.27	21.46

Table 3: General statistics for essays in the corpus

Table 4 shows the distribution of RCs for different L1 backgrounds, their rate and percentage of occurrence with respect to sentences. RCs are found to be a high-frequency feature for Italian: RCs occur in every 3.23 sentences, or 30.93% of the sentences contain an RC. By contrast, RCs occur least frequently for Urdu (only in every 11.81 sentences or in 8.47% of all sentences).

L1	# RCs	# sentences	rate	%
Finnish	187	945	5.05	19.79
Italian	202	653	3.23	30.93
Polish	163	796	4.88	20.48
Swedish	147	705	4.80	20.85
Turkish	137	942	6.88	14.54
Urdu	88	1039	11.81	8.47
TOTAL	924	5080	5.50	18.19

Table 4: Distribution of RCs

5 Results for referent functions

We begin with presenting the distribution of referent functions in the corpus, as shown in Table 5[22]. Overall, direct objects in the matrix clauses are found to be relativised most frequently in the RCs (32.25%, in the rightmost column), followed by adjuncts, predicative complements, and subjects. By contrast, the least frequently relativised items are (matrix) clauses and indirect objects.

The pattern, however, does not apply strictly on individual L1 backgrounds. For example, for Polish the pattern is less strongly pronounced (with the scores for the categories being close to each other), or for Swedish, adjuncts (instead of direct objects) in the matrix clauses are relativised most often, or for Urdu, subjects and predicative complements score higher than adjuncts.

Next, we examine the co-occurrence of referent functions and other RC features. First, Table 6 presents the distribution (in percentages) of the RM types for referent functions[23]. Overall, across all L1s *wh*-words (e.g., *which*, *who*, *whose*) constitute the most common RM type in the RCs, regardless of the referent functions. The students with L1 Urdu are found to use *wh*-words almost exclusively for RMs. This partially holds for Italian (only with the `subj` feature) and Polish (only with the `pred-comp` feature). Turkish almost never uses `zero` (bare relatives).

Second, Table 7 shows the co-occurrence of referent functions and marker functions. First of all, for all L1s the relativised items most often serve as the subject of the RCs, regardless of the referent functions. For specific L1s, some patterns are observed:

1. `subj` ∼ `subj`: When the referent is the subject in the matrix clause, the RM also tends to be the subject of the RC. This applies almost exclusively for Swedish, Turkish, and Urdu.
2. `dir-obj` ∼ `dir-obj`: When the referent is the direct object, the RM serves more often as a direct object (after the subject).
3. `pred-comp/adjunct` ∼ `dir-obj`: When the referent is a predicative complement or an adjunct, the RM is more often as an adjunct rather than a direct object (after the subject).

Third, we examine the co-occurrence of referent functions and referent types (e.g, human, abstract entity) in Table 8[24]. Overall, for all L1s the most common referent type is `abstract entity`, irrespective of the referent functions. However, the difference between the preference for `human` and `abstract entity` is less clear when the referent serves as the subject in the matrix clause. In fact, `human` outscores `abstract entity` in such a configuration for Polish, Turkish, and Urdu. By contrast, `non-human` (concrete) entities are rarely relativised in the RCs.

Finally, the co-occurrence of referent functions and restrictiveness in Table 9 shows that L2 English

[21] The official ICLE data collection instructions stipulate ca. 600 words per essay.

[22] The occurrence of fewer than 5 tokens for a category was excluded from all the tables.

[23] The `indir-obj` and `clause` features are excluded from the tables due to low frequency.

[24] The `proposition` feature was excluded from the table due to its low frequency.

type	Finnish	Italian	Polish	Swedish	Turkish	Urdu	avg
subj	32 (17.11%)	34 (16.83%)	34 (20.86%)	25 (17.01%)	22 (16.06%)	21 (23.86%)	168 (18.18%)
dir-obj	61 (32.62%)	69 (34.16%)	41 (25.15%)	49 (33.33%)	50 (36.50%)	28 (31.82%)	298 (32.25%)
indir-obj	-	-	-	-	-	-	14 (1.52%)
pred-comp	43 (22.99%)	38 (18.81%)	31 (19.02%)	15 (10.20%)	25 (18.25%)	18 (20.45%)	170 (18.40%)
adjunct	42 (22.46%)	57 (28.22%)	36 (22.09%)	51 (34.69%)	29 (21.17%)	13 (14.77%)	228 (24.68%)
clause	7 (3.74%)	-	17 (10.43%)	-	9 (6.57%)	7 (7.95%)	46 (4.98%)
TOTAL	187	202	163	147	137	88	924

Table 5: Distribution of referent functions

type	RM	Finnish	Italian	Polish	Swedish	Turkish	Urdu	avg
subj	*that*	18.75	-	17.65	24.00	27.27	-	18.45
	wh-word	56.25	82.35	73.53	52.00	63.54	80.95	68.45
	zero	25.00	-	-	24.00	-	-	13.10
dir-obj	*that*	36.07	24.64	17.07	34.69	36.00	-	28.19
	wh-word	49.18	56.52	68.29	44.90	58.00	82.14	57.38
	zero	14.75	18.84	14.63	20.41	-	-	14.43
pred-comp	*that*	30.23	18.42	-	60.00	28.00	-	25.29
	wh-word	41.86	71.05	77.42	33.33	60.00	77.78	60.59
	zero	27.91	-	-	-	-	-	14.12
adjunct	*that*	26.19	17.54	-	25.49	31.03	-	20.61
	wh-word	54.76	66.67	72.22	54.90	65.51	76.92	63.16
	zero	19.05	15.79	22.22	19.61	-	-	16.23

Table 6: Co-occurrence of RMs and referent functions

type	m-function	Finnish	Italian	Polish	Swedish	Turkish	Urdu	avg
subj	subj	68.75	79.41	82.35	68.00	77.27	90.48	77.38
	dir-obj	18.75	-	14.71	-	-	-	12.50
	adjunct	-	14.71	-	-	-	-	8.93
dir-obj	subj	57.38	55.07	60.98	59.18	64.00	64.29	59.40
	dir-obj	26.23	28.99	17.07	26.53	22.00	17.86	24.16
	adjunct	14.75	15.94	19.51	12.24	10.00	17.86	14.77
pred-comp	subj	51.16	63.16	54.84	66.67	52.00	66.67	57.65
	dir-obj	30.23	13.16	19.35	-	16.00	-	18.82
	adjunct	18.60	18.42	25.81	-	32.00	33.33	22.35
adjunct	subj	57.14	59.65	50.00	52.94	65.52	76.92	57.89
	dir-obj	-	17.54	22.22	17.65	20.69	-	15.79
	adjunct	35.71	15.79	22.22	25.49	-	-	22.81

Table 7: Co-occurrence of marker functions and referent functions

users, regardless of their L1s, use integrated RCs more often than supplementary RCs. The pattern is more strongly pronounced for Finnish, Swedish, Turkish, and Urdu when the referent is the subject. This also holds true for Swedish, Turkish, and Urdu, when the referent is the predicative complement.

6 Discussion

In the ICLE-RC, the students (advanced L2 learners of English) were found to relativise all major constituents in the matrix clause in the RCs, but with varying degrees: direct objects > adjunct > predicative complement / subject > (matrix) clause > indirect object (in Table 5). This order is, however, not corroborated by NPAH (Keenan and Com-

type	ref-type	Finnish	Italian	Polish	Swedish	Turkish	Urdu	avg
subj	human	37.50	44.12	52.94	32.00	54.55	61.90	45.43
	non-human	-	-	-	-	-	-	2.98
	abstract	56.25	50.00	41.12	68.00	45.45	38.10	50.00
dir-obj	human	19.67	15.94	-	18.37	26.00	-	17.45
	non-human	-	14.49	24.39	18.37	-	-	10.74
	abstract	78.69	65.22	65.85	63.27	74.00	82.14	70.81
pred-comp	human	13.95	21.05	16.13	33.33	20.00	-	18.24
	non-human	-	13.16	-	-	-	-	7.65
	abstract	79.07	65.79	77.42	53.33	76.00	83.33	73.53
adjunct	human	14.29	28.07	33.33	15.69	20.69	38.46	23.25
	non-human	-	-	-	17.65	-	-	7.46
	abstract	80.95	66.67	61.11	66.67	72.41	53.85	68.42

Table 8: Co-occurrence of referent types and referent functions

type	restrictiveness	Finnish	Italian	Polish	Swedish	Turkish	Urdu	avg
subj	integrated	87.50	58.82	82.35	88.00	100.00	85.71	82.14
	supplementary	-	41.18	17.65	-	-	-	17.86
dir-obj	integrated	63.93	71.01	60.98	77.55	78.00	64.29	69.80
	supplementary	36.07	28.99	39.02	22.45	22.00	35.71	30.20
pred-comp	integrated	81.40	47.37	61.29	80.00	88.00	72.22	70.00
	supplementary	18.60	52.63	38.71	-	-	-	30.00
adjunct	integrated	73.81	57.89	61.11	74.51	68.97	69.23	67.11
	supplementary	26.19	42.11	38.89	25.49	31.03	-	32.89

Table 9: Co-occurrence of integrated/supplementary RCs and referent functions

rie, 1977). We find this quite intriguing, and call for a closer scrutiny of a larger variety of learner English. We also observed deviations from this pattern for Swedish (adjunct > direct object > subject > predicative complement) and Urdu (direct object > subject > predicative complement > adjunct). This raises an important question: Do these specific orders for constituents originate from the ways RCs are structured in those languages, or do they show influence of prior (institutionalised) learning? Unfortunately, as studies on referent functions are not abundant, we cannot directly compare our results to previous research, and we thus leave these inquiries for further investigation.

Some other patterns, however, can potentially be explained with reference to the ways RCs function in different L1s. For example, the students with L1 Urdu overwhelmingly used an overt RM, particularly *wh*-words (in Table 6). A scrutiny of the Urdu grammar reveals that (finite) RCs in Urdu, like in many other Indo-Aryan languages, are introduced with a correlative construction: a demonstrative pronoun + a relative pronoun (Srivastav, 1991; Bhatt, 2003). This is illustrated by the *vo-jo* pair in (26), taken from Butt et al. (2007, p.113).

(26)

vo	*laṛki*	***[jo***
that.Dem.	girl.F.Sg.Nom	which.Rel.

k^haṛi	*hɛ]*	*lɑmbi*
stand-Perf.F.Sg.	be.Pres.3.Sg	tall.F.Sg.

hɛ
be.Pres.3.Sg.

'The girl who is standing is tall.'

This explicit (double-)marking of RCs might have a direct influence on the Urdu students for not preferring the use of bare-relatives in English.

The same reasoning apparently fails to apply to Turkish, however. Turkish does not employ an overt *wh*-element or complementiser to introduce RCs; rather, RCs are marked morphologically by certain particles (suffixes), as shown in (27), taken from Kornfilt (1997, p.29).[25]

[25] In fact, it has traditionally been argued that Turkish lack genuine RCs, and have only deverbal adjectives: *a running child* instead of *a child who is running* (Kornfilt, 2000, p.123).

(27)

[geçen	*yaz*	*ada-da*	*ben-i*
last	summer	island-Loc.	I.Acc.

*gör-**en**]*	*kişi-ler*
see.Part.	person.Pl.

'The people who saw me on the island last summer'

Like Urdu, had we assumed an L1 effect of Turkish on the structuring of English RCs, we would have expected that the Turkish students would use mostly bare-relatives in English. Yet, we find counter-evidence in our data: The Turkish students (like Urdu students) have almost always used an overt RM for RCs in English. Slobin (1986) argues that Turkish RCs are not readily isolable since they are synthetic and even noncanonical to a Turkish clause. Furthermore, the processing of RCs in Turkish necessitates the use of more demanding strategies by children acquiring the language. By contrast, English RCs are analytic and canonical to an English clause. Based on this, we speculate that the Turkish students, when producing RCs in English, might have resorted to using the more distinguishable, canonical English RC structures involving the use of an overt RM. Alternatively, it might also be the case that since Turkish RCs are always marked, albeit by a particle, the Turkish students chose to always mark the English RCs by an overt RM rather than leave them unmarked (i.e., use bare-relatives). In any of these cases (and beyond), we believe that these conflicting results have important implications for research on the competing roles between L1 influence and the efficacy and success of L2 instructions, and require further exploration.

Next, the distribution of the marker functions (in Table 7) shows a clear ordering: subject > direct object / adjunct. This is validated by previous research (e.g., Tavakoli, 2013). Furthermore, the co-occurrence of referent functions and marker functions, however, shows some interesting patterns (see previous section). These patterns may well be determined based on the product of the relative complexity of each of the functions (Hundt et al., 2012), the distance between referent-heads and RM (Tagliamonte et al., 2005), or the level of RC-embedding (Karlsson, 2007). This, we feel, falls beyond the scope of the present study, and we intend to investigate it further in our future work.

For referent types, Fox and Thompson (1990) found that non-human subject heads in the matrix clause tend to co-occur with the objects in the RCs, and also that non-human object heads in the matrix clause do not tend to co-occur with the objects in the RC[26]. This is partially corroborated by our data, as we found evidence only for the second claim, but counter-evidence for the first one. The distribution of the relevant categories is provided in Table 10.

Finally, the prevalence of integrated RCs in the ICLE-RC indicates that the RCs are used more often as an integral part of the referent NP rather than providing additional information or commentary about it. This implies that L2 English learners use RCs more as a syntactic device than a discourse one (i.e., RCs as discourse segments).

7 Conclusions and outlook

RC-referents in the ICLE-RC show variation for their syntactic functions across different L1 backgrounds. The variation seems even greater and multifarious when their co-occurrence with other RC-features is taken into account. In our future work, we would conduct a thorough examination of the RC-related grammar of each L1, and test our findings against them to see whether any cross-linguistic factors influence the patterning of referent functions in the English RCs.

The ICLE-RC is now in the post-production stage, and will soon be published as an open-access resource. Our future work would include expanding the size and coverage of the corpus by adding more texts for the existing six L1s as well as incorporating texts from other L1 backgrounds (from the ICLE), representing new (sub-)language families; e.g., Cantonese (Sino-Tibetan), Dutch (West Germanic), Greek (Hellenic), Japanese (Japonic), Farsi (Indo-Iranian), Russian (Slavic), Tswana (Bantu). This would facilitate large-scale studies on referent functions and many other RC-related phenomena.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback on the first submission version of the paper.

References

J.C. Acuña Fariña. 2000. Reduced relatives and apposition. *Australian Journal of Linguistics*, 20(1):5–22.

[26] This study is, however, not directly comparable to ours as it examined the use of RCs in (non-learner) conversations.

ref-type	r-function	m-function	#
non-human and abstract entity	subj	subj	53 (8.15%)
		dir-obj	20 (3.08%)
	dir-obj	subj	126 (19.39%)
		dir-obj	69 (10.62%)
TOTAL			650

Table 10: Co-occurrence of non-human, referent functions and marker functions

C. Bache and L.K. Jakobsen. 1980. On the distinction between restrictive and non-restrictive relative clauses in modern english. *Lingua*, 52(3):243–267.

R. Bhatt. 2003. Locality in correlatives. *Natural Language & Linguistic Theory*, 21:485–541.

D. Biber, S. Johansson, G. Leech, S. Conrad, and E. Finegan. 1999. *Longman Grammar of Spoken and Written English*. Pearson Education Limited.

S. Brandt, E. Kidd, E. Lieven, and M. Tomasello. 2009. The discourse bases of relativization: An investigation of young german and english-speaking children's comprehension of relative clauses. *Cognitive Linguistics*, 20(3):539–570.

M. Butt, T.H. King, and S. Roth. 2007. Urdu Correlatives: Theoretical and Implementational Issues. In *Proceedings of LFG '07 Conference*, pages 107–127, Stanford, California. CSLI Publications.

B. Comrie. 1998. Rethinking the typology of relative clauses. *Language Design*, 1:59–86.

F. Cornish. 2018. Revisiting the system of English relative clauses: structure, semantics, discourse functionality. *English Language and Linguistics*, 22:431–456.

D. Das, I. Czerniak, and P. Bourgonje. to appear. ICLE-RC: International Corpus of Learner English for Relative Clauses. In *Proceedings of the 19th Linguistic Annotation Workshop*.

H. Diessel and M. Tomasello. 2005. A New Look at the Acquisition of Relative Clauses. *Natural Language & Linguistic Theory*, 81(4):882–906.

C. Doughty. 1991. Second Language Instruction Does Make a Difference: Evidence from an Empirical Study of SL Relativization. *Studies in Second Language Acquisition*, 13(4):431–469.

M.S.A. Fajri and V. Okwar. 2020. Exploring a Diachronic Change in the Use of English Relative Clauses: A Corpus-Based Study and Its Implication for Pedagogy. *SAGE Open*, 10(4).

B.A Fox and S.A. Thompson. 1990. A Discourse Explanation of the Grammar of Relative Clauses in English Conversation. *Language*, 66(2):297–316.

H. Goad, N.B. Guzzo, and L. White. 2021. Parsing Ambiguous Relative Clauses in L2 English: Learner Sensitivity to Prosodic Cues. *Studies in Second Language Acquisition*, 43(1):83–108.

S. Granger. 1998. The computer learner corpus: A versatile new source of data for sla research. In S. Granger, editor, *Learner English on Computer*, pages 3–18. Addison Wesley Longman, London & New York.

S. Granger, M. Dupont, F. Meunier, H. Naets, and M. Paquot. 2020. The International Corpus of Learner English. Version 3.

A. Grosu. 2012. Towards a More Articulated Typology of Internally Headed Relative Constructions: The Semantics Connection. *Language and Linguistics Compass*, 6(7):447–476.

L. Hinrichs, B. Szmrecsanyi, and A. Bohmann. 2015. "WHICH"-HUNTING AND THE STANDARD ENGLISH RELATIVE CLAUSE. *Language*, 91(4):806–836.

R. Huddleston and G.K. Pullum. 2002. *The Cambridge grammar of the English language*. CUP, Cambridge, UK.

M. Hundt, D. Denison, and G. Schneider. 2012. Relative complexity in scientific discourse. *English language and linguistics*, 16(2):209–240.

S. Ishikawa. 2023. *The ICNALE Guide: An Introduction to a Learner Corpus Study on Asian Learners' L2 English*. Routledge.

F. Karlsson. 2007. Constraints on multiple center-embedding of clauses. *Journal of Linguistics*, 43(2):365–392.

E. Keenan and B. Comrie. 1977. Noun Phrase Accessibility Hierarchy and Universal Grammar. *Linguistic Inquiry*, 8:63–99.

E. Kidd. 2011. *The Acquisition of Relative Clauses: Processing, typology and function*. Benjamins, Amsterdam.

J. Kornfilt. 1997. On the Syntax and Morphology of Relative Clauses in Turkish. *Dilbilim Araştırmaları Dergisi*, 8:24–51.

J. Kornfilt. 2000. Some syntactic and morphological properties of relative clauses in Turkish. In *The Syntax of Relative Clauses*, pages 121–159. John Benjamins, Amsterdam/Philadelphia.

R. Landis and G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174.

G. Leech, M. Hundt, C. Mair, and N.I. Smith. 2009. *Change in contemporary English: A grammatical study*. Cambridge University Press.

G. McKoon and R. Ratcliff. 2003. Meaning Through Syntax: Language Comprehension and the Reduced Relative Clause Construction. *Psychological review*, 110(3):490–525.

M. O'Donnell. 2008. The UAM CorpusTool: Software for corpus annotation and exploration. In Carmen M.. Bretones Callejas, editor, *Applied Linguistics Now: Understanding Language and Mind / La Lingüística Aplicada Hoy: Comprendiendo el Lenguaje y la Mente*, pages 1433–1447. Almería, Universidad de Almería.

A. Pereltsvaig. 2023. *Languages of the World: An Introduction*, 4th edition. Cambridge University Press.

F. Reali and M.H. Christiansen. 2007. Processing of relative clauses is made easier by frequency of occurrence. *Journal of Memory and Language*, 53:1–23.

D.I. Slobin. 1986. The acquisition and use of Relative Clauses in Turkic and Indo-European Languages. In *Studies in Turkish Linguistics*, page 277–298. John Benjamins, Amsterdam.

V. Srivastav. 1991. The Syntax and Semantics of Correlatives. *Natural Language and Linguistic Theory*, 9(4):637–686.

C. Suárez-Gómez. 2006. *Relativization in Early English (950–1050): The Position of Relative Clauses*. Peter Lang.

C. Suárez-Gómez. 2015. The places where English is spoken: adverbial relative clauses in World Englishes. *World Englishes*, 34(4):620–635.

S. Tagliamonte, J. Smith, and H. Lawrence. 2005. No taming the vernacular! Insights from the relatives in northern Britain. *Language Variation and Change*, 17:75–112.

H. Tavakoli. 2013. *A dictionary of language acquisition: A comprehensive overview of key terms in first and second language acquisition*. Rahnama, Tehran.

D. Weichmann. 2015. *Understanding relative clauses: A usagebased view on the processing of complex constructions*. De Gruyter Mouton.

A Appendix

language	institution	gender	# essays
Finnish (Uralic)	University of Helsinki	F	4
		M	4
	University of Joensuu (now UEF)	F	4
		M	4
	University of Jyväskylä	F	4
		M	4
Italian (Romance)	University of Bergamo	F	6
		M	2
	Sapienza University of Rome	F	4
		M	4
	University of Turin	F	4
		M	4
Polish (Slavic)	Maria Curie-Skłodowska University	F	8
		M	0
	Adam Mickiewicz University	F	4
		M	4
	University of Silesia in Katowice	F	8
		M	0
Swedish (Germanic)	University of Gothenburg	F	4
		M	4
	Lund University	F	4
		M	4
	Växjö University	F	6
		M	2
Turkish (Turkic)	Mersin University	F	4
		M	8
	University of Mustafa Kemal	F	2
		M	2
	University of Çukurova	F	8
		M	0
Urdu (Indo-Aryan)	GC University Faisalabad	F	4
		M	8
	Govt College for Women Jhang	F	2
		M	2
	Lahore College for women university	F	8
		M	0
TOTAL			144

Table 11: Distribution of the essays in the ICLE-RC

RC annotation feature			
level 1	**level 2**	**level 3**	**level 4**
RM	that		
	wh-word	*which*, *who*, *whose*, etc.	
	zero		
referent function	subject	subj-head-n	
		in-subj-comp	
		in-subj-adjunct	
	direct obj	dir-obj-head-n	
		in-dir-obj-comp	
		in-dir-obj-adjunct	
	indirect obj	indir-obj-head-n	
		in-indir-obj-comp	
		in-indir-obj-adjunct	
	predicative complement	pred-comp-np	pred-comp-head-n
			in-pred-comp-np-comp
			in-pred-comp-np-adjunct
		pred-comp-adjp	pred-comp-head-adj
			in-pred-comp-adjp-comp
			in-pred-comp-adjp-adjunct
		pred-comp-pp	pred-comp-head-p
			in-pred-comp-pp-comp
	adjunct	adjunct	
		in-adjunct	
	clause		
marker function	subject		
	direct obj		
	Indirect obj		
	predicative complement	pred-comp-full	
		in-pred-comp	
	gen-subj-det		
	predicate		
	aux-comp		
	head-to-inf-vp		
	adjunct		
embedding	yes		
	no		
extraposition	yes		
	no		
ref type	entity	human	
		non-human	
	abstract		
	proposition		
restrictiveness	integrated		
	supplementary		

Table 12: Taxonomy of features for RC annotation

The sentence in which the RC features are to be annotated: Unfortunately, life is not a situation comedy ***where*** *every problem is happily solved.* [Italian; ITTO-1002]		
meta-features	L1	Italian
	institution	University of Turin
	gender	female
RC features	RM	wh-word → *where*
	referent function	pred-comp → pred-comp-np → pred-comp-head-n
	marker function	adjunct
	embedding	no
	extraposition	no
	referent type	abstract entity
	restrictiveness	integrated

Table 13: Example of RC annotation

An intonosyntactic treebank for spoken French:
What is new with Rhapsodie?

María Paz Botero-Garcia[1] **Sylvain Kahane**[1,3] **Emmett Strickland**[1]
Bruno Guillaume[2] **Anne Lacheret-Dujour**[1]
[1]MoDyCo, Université Paris Nanterre & CNRS [2]Université de Lorraine, CNRS, Inria, LORIA [3]Institut Universitaire de France

Abstract

This paper presents a new format of the Rhapsodie Treebank, which contains both syntactic and prosodic annotations, offering a comprehensive dataset for the study of spoken French. This integrated format allow us for complex multilevel queries and open the way for the extraction of intonosyntactic studies.

1 Introduction

The Rhapsodie Treebank is the outcome of the French National Research Agency (ANR) project Rhapsodie, which began in 2008. It is the fruit of years of work by a group of French researchers who collected 3 hours and 10 minutes of spoken French audio, transcribed it, analyzed it, and developed a multi-level annotation scheme (wich involves syntax and prosody) that is reproducible and allows for the study of the syntax-prosody interface in French (Lacheret-Dujour et al., 2019b).

The main interest of this corpus, in addition to its multilevel annotation, lies in the richness of its metadata. It is composed of 30 monologues and 27 dialogues produced with different communicative goals, which may belong to public or private social contexts and can be spontaneous, semi-spontaneous, or planned, with varying degrees of interactivity.

In this paper, we propose to implement and expand upon the methodology introduced in Strickland et al. (2024) for Naija, or Nigerian Pidgin, to provide a new version of the Rhapsodie treebank where the different annotation layers (morphosyntax and prosody) are represented in a unified structure. The main benefit is that this version allows for a more in-depth study of the interaction between syntax and prosody. This is illustrated in the paper cited above on Naija.

2 Combining syntax and prosody in one single format

In 2024, the intonosyntactic treebank for the Naija Strickland et al. (2024) introduced for the first time a format in which every node annotated with syntactic information, (i. e., each token), is associated with child nodes corresponding to its constituent syllables, which are annotated with automatically extracted prosodic information. Since the Rhapsodie project had already been manually annotated at both the syntactic and prosodic levels years earlier, this development represented a valuable opportunity to adapt the original Rhapsodie corpus to the new format, while preserving the original manual annotations and incorporating new ones.

The main difference between the Rhapsodie intonosyntactic treebank and the Naija intonosyntactic treebank is the presence of micro- and macro-syntactic annotations and extra prosodic annotations at the token level and not just at the syllable level, like the token's position within an intonative period, metrical foot, rhythmic group or an intonational package, that will be explained later. The inclusion of prosodic information altered the structure of the dependency tree, as two tokens may share the same syllable (fused syllable). Their corresponding subtokens are therefore connected accordingly, as shown in Figure 1.

The corpus update involved the integration of syntactic and prosodic information into a single unified CoNLL-U format to facilitate its use. The original Rhapsodie Treebank is composed of approximately 33,000 tokens, for which the multi-level annotations were distributed in various formats.[1] These included WAV/MP3 files for the audio, TXT files for the transcription, tabular formats for micro- and macro-syntactic annotations, pitch format for acoustic analysis, and XML, TextGrid, and tabular formats for prosodic annotations. Metadata was

[1] https://rhapsodie.modyco.fr

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 111–118
August 28-29, 2025 ©2025 Association for Computational Linguistics

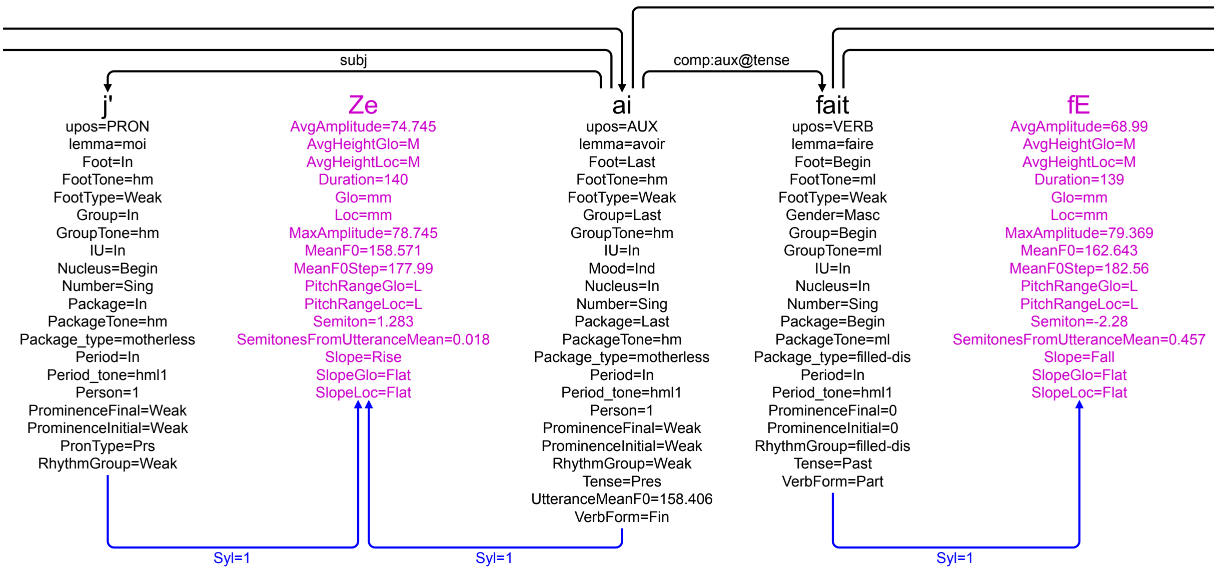


Figure 1: Example with prosodic annotations. Syllable *Ze* is fused, forming a graph structure in the sequence *et puis j'ai fait mes études au lycée, euh, de Mulhouse* ('and then I did my studies at the high school, uh, in Mulhouse'), [Rhap-D2004].

provided in both HTML and XML formats.

Since the corpus was developed over the years, with numerous reseachers involved, differences emerged in the corpus segmentations and in the alignment between each annotation. As a result, updating the corpus represented a significant challenge. The work was divided in two main steps. First, the annotations from the original Rhapsodie treebank were grouped, aligned, extracted, added and normalized. Second, automatic annotations were obtainet with the work of Strickland et al. (2023), added and normalized.

3 Integration of existing annotations

3.1 Syntax

Regarding the syntactic information provided by the original version of the corpus, in addition to a morphosyntactic analysis for word segmentation and lemmatization, the original version of Rhapsodie used its own annotation scheme, inspired by dependency syntax and the syntax of spoken corpora, in which syntactic boundaries are evaluated differently than in written corpora, relying on macrosyntax (Gerdes and Kahane, 2017).

The main difference between Rhapsodie's annotation scheme, Universal dependencies (UD) and Surface-Syntactic Universal Dependencies (SUD) is that in Rhapsodie, macrosyntax and microsyntax were annotated separately, neither of which involves prosodic criteria, although both interplay in complex ways in spoken language. Macrosyntax refers to syntactic cohesion ensured by the illocutionary act and microsyntax refers to syntactic cohesion based on government relations. The latter is encoded in a dependency tree, where a single head, which is not governed itself, projects governability onto the other lexemes (Lacheret-Dujour et al., 2019b).

The samples of the corpus are macrosyntactically segmented into groups of syntactic constituents, major syntactic units, that perform the same illocutionary act, called Illocutionary Units (IUs). Illocutionary acts include assertion, induction, interrogation, and exclamation. We can say that they perform the same illocutionary act because they can be placed under the scope of a verb that makes explicit the force of the illocutionary unit. In the first example, each IU boundary is marked with a double slash ("//").

(1) *L2 donc < moi < "ben" { je vais | {*
L2 so < me < "well" { I'm going | {
je | je } prends le mét~| je prends le
I | I } take the met~| I take the
métro } le matin "bon" jusqu'au Palais
metro } in the morning "okay" up to the Palais
Royal //+ L1 à quelle heure //
Royal //+ L1 at what time //
"excusez-moi" // [Rhap_D0001]
"excuse me" //

From a microsyntactic perspective, a Government Unit (GU) consists of all the lexemes that form the dependency graph. Even though *je prends* *le métro le matin bon jusqu'au Palais Royal* and *à quelle heure* are two different IUs and two different turns of speech, they belong to the same GU (Kahane et al., 2019). The second illocutionary unit is governed by the first one. The dependency relationship between the two is categorized as *mod* in SUD.[2]. It can be queried on https://universal.grew.fr/?corpus=SUD_French-Rhapsodie_db

Based on this analysis, the project established the following annotations for macrosyntax. Each annotation describes the token's inclusion within a constituent or IU and its place, in BILU format (Begin, Inside, Last, Unique).

In the second example, the token *nous* 'we' is annotated as `IU=Begin` and *déja* 'first' as `IU=Last`.

(2) *nous < dans le quartier <+* ***on n'a***
we < in the neighborhood <+ **we don't**
on n'a pas de lycée *> déjà //*
we don't have any high schools > first //
`[Rhap_D0004]`

Each IU has a **nucleus**, which is the autonomous constituent that makes clear what kind of act the speaker is performing. In this example, the token *on* 'we' is annotated as `Nucleus=Begin` and *lycée* 'school' as `Nucleus=Last`.

Other constituents inside the IU that cannot be the scope of a predicate without the nucleus, because they depend illocutionarily on it, are considered ad-nuclei. In Rhapsodie, they are classified as pre-nuclei (on the left of the nucleus), in-nuclei (within the nucleus), and post-nuclei (on the right of the nucleus). In the second example, *dans le quartier* 'in the neighborhood', *dans* is annotated as `Prenucleus=Begin` and *quartier* as `Prenucleus=Last`.

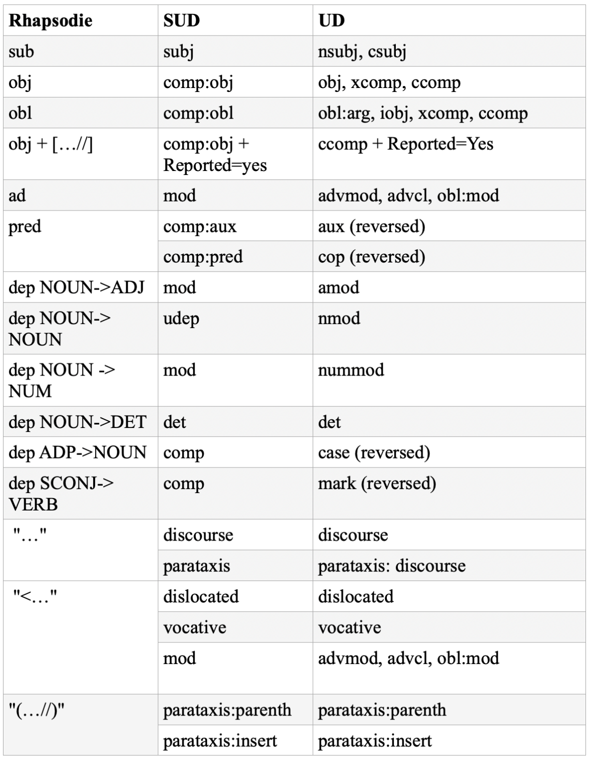
A **graft** is when the speaker does not find the good denomination and graft an IU where a proper noun was expected. In the third example, *je crois que c'est une ancienne caserne, je crois* 'I think it is an ancient barracks, I think' is used instead of *une ancienne caserne* 'an ancient barracks'. All tokens of the graft bear a feature IU_graft, with a BILU value, and the root of the graft (the first *crois*) has an additional feature Graft=Yes.

(3) *vous t~ vous suivez la ligne du tram qui*
you t~ you follow the tram line which
passe vers la &
goes toward the &
[je crois que c'est une ancienne caserne "je crois" //
[I think it is an ancient barracks "I think" //]
] // `[Rhap_M0003]`
//

The label **AssociatedNucleus** appears when a GU shares distributional properties with nuclei, but carries a weak illocutionary force, as *je pense* 'I think' in the fourth example.

(4) *ça < c'est le problème de Paris*
that < that's the problem of Paris
"je pense" // `[Rhap_D0004]`
"I think" //

There are also differences between the Rhapsodie annotation scheme and the UD and SUD annotation schemes in the naming of dependency relations at the microsyntactic level, as shown in Figure 2. The updated version of the corpus was produced from a recent version annotated according to the SUD scheme which was developed on the basis of the Rhapsodie format, enriching it to allow for conversion into UD. The segmentation in IUs is preserved and the previously mentioned macrosyntactic annotations were added.



Rhapsodie	**SUD**	**UD**
sub	subj	nsubj, csubj
obj	comp:obj	obj, xcomp, ccomp
obl	comp:obl	obl:arg, iobj, xcomp, ccomp
obj + […//]	comp:obj + Reported=yes	ccomp + Reported=Yes
ad	mod	advmod, advcl, obl:mod
pred	comp:aux	aux (reversed)
	comp:pred	cop (reversed)
dep NOUN->ADJ	mod	amod
dep NOUN-> NOUN	udep	nmod
dep NOUN -> NUM	mod	nummod
dep NOUN->DET	det	det
dep ADP->NOUN	comp	case (reversed)
dep SCONJ-> VERB	comp	mark (reversed)
"…"	discourse	discourse
	parataxis	parataxis: discourse
"<…"	dislocated	dislocated
	vocative	vocative
	mod	advmod, advcl, obl:mod
"(…//)"	parataxis:parenth	parataxis:parenth
	parataxis:insert	parataxis:insert

Figure 2: Correspondence between the Rhapsodie, SUD, and UD annotation schemes.

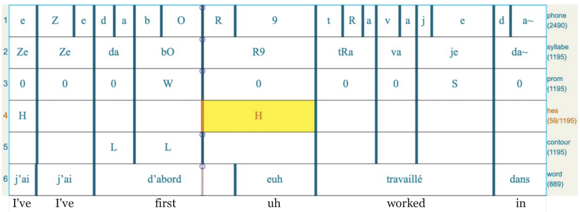
3.2 Prosody

Regarding the prosodic annotation in Rhapsodie, it follows a data-driven approach which has been divided into three stages: prominence and disfluency

[2]This can be verified in this query, which shows the *mod* relation between illocutionary units in the new dependency-based version of the Rhapsodie treebanks of GREW-MATCH

annotation, segmentation into maximal prosodic units called Intonational Periods (IPEs), and intonational annotation relative to the intonation contour. (Lacheret-Dujour, 2019)

A syllable is considered **prominent** if it is perceptually more salient than its surrounding context. It can be annotated as `Weak`, `Strong`, or `0` if it is not prominent (Avanzi et al., 2019). If the syllable belongs to the filler *euh* 'uh' or exhibits features such as extra lengthening, infra-low register, or creaky voice, it is marked with `H`, indicating the presence of a **hesitation** (see Figure 3).



1	e	Z	e	d	a	b	O	R	9	t	R	a	v	a	j	e	d	a~	phone (2490)
2	Ze	Ze		da		bO		R9		tRa			va		je		da~		syllabe (1195)
3	0	0		0		W		0		0			0		S		0		prom (1195)
4	H							H											hes (59/1195)
5				L		L													contour (1195)
6	j'ai	j'ai		d'abord					euh	travaillé							dans		word (889)
	I've	I've		first					uh	worked							in		

Figure 3: Original TextGrid with annotations for period, word, contour, prominence, syllable, and phonetic transcription for the sequence *j'ai j'ai d'abord euh travaillé dans* 'I've I've first uh worked in' `[Rhap_D0005]`.

Segmentation into Intonative Periods (IPEs) is based on perceptual and acoustic cues. It is important to note that segmentation into IPEs does not necessarily align with that of IUs, since the IPE identification does not involve syntactic criteria.[3] It occurs when a silence of 300 milliseconds, with an absence of a filler contiguous to the pause, is detected and associated to a marked terminal contour before the pause and a melodic resetting after the pause. The detection of a speech turn is also associated to the end of a period (Lacheret-Dujour and Victorri, 2019). The token's position within an IPE is marked in BILU format too.

The period represents the root of the prosodic tree which is articulated around 3 levels of constituency from the bottom up: metrical foot, rhythmic group and intonation package as shown. Every time there is a non-disfluent but prominent syllable within an IPE, the end of a **metrical foot** (MF) is marked. In other words, a metrical foot can be composed of non-prominent syllables followed by a prominent syllable, also called the Right Head of the Foot (RHF). The prominence of the RHF determines the label of the foot as either strong or weak. A **rhythmic group** (RG) boundary is marked when a RHF (Right Head of the Foot) coincides with the final syllable of a token. When rhythmic groups occur in succession, an **intonation package** (IPA) is marked by the first group that carries a strong prominence (see Figure 4).

Sy	re	po~	sa	sHi	vRa	vE	ke	lEn	S@	va	lije
Prom	S		W	S				W	W		S
MF	re		po~ sa	sHi	vRavEkelEn				S@	valije	
RG	**réponse à**			**suivre**	**avec Hélène**				**Chevalier**		
IPA	**réponse à suivre**				**avec Hélène Chevalier**						

Figure 4: Example of a segmentation in MF, RG and IPA, where four rhytmic groups form two intonation packages in the sequence *réponse à suivre avec Hélène Chevalier* 'response to follow with HC', [Rhap-M2006]. Extracted from Lacheret-Dujour et al. (2019a).

3.3 Integration of new features

The main contribution of the treebank format introduced in Strickland et al. (2024) to ours lies in enabling quantitative studies based on acoustic analysis, both segmentally and suprasegmentally. This work used SLAM 3 (Strickland et al., 2023), the latest version of the SLAM prosodic modeling software that generates discrete labels from continuous F0 contours, which are otherwise difficult to manipulate. We also continued the Naija corpus's annotation of continuous features extracted directly from the .TextGrid and .PitchTier files of an audio, such as the duration of each syllable or `UtteranceMeanF0`, which indicates the mean F0 of each sequence in the corpus. It should be noted that our data is macrosyntactically segmented into IUs, and therefore, this annotation appears at the token that is the root of every IU.

Information related to the pitch contour is extracted from the raw pitch curves with SLAM 3, both at the global level (token) and at the local level (syllable). The pitch onset and offset of each selected segment are considered to generate a discrete label, along with its most prominent point. These discrete labels can take the values very low (L), low (l), medium (m), high (h), or very high (H). The system applies a glissando threshold formula to ensure that only pitch changes perceptible over time are taken into account (Strickland et al., 2023). It then assigns one `Glo` and one `Loc` label per syllable, corresponding respectively to `[pitch's start - pitch's end - pitch's most salient point of the contour - syllabic tier in which this prominence occurs]` like this **`[mlh1]`**. In this label, the pitch starts at a medium level (m), ends at a low level(l), with a salient high peak (h), all occurring in the first tier of the syllable (1). The

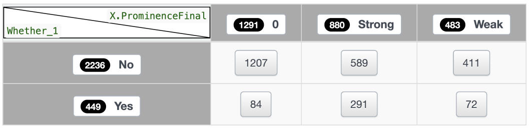
[3] This segmentation was performed using the Analor tool (Avanzi et al., 2008) and was manually verified by an expert.

full version of the annotations that were used for the Rhapsodie's new format is available in the appendix, items (27–41).

4 Use of this resource

The updated corpus is still undergoing refinement, but it is already available on GREW-MATCH.[4] For most samples in the corpus, users can visualize a dependency tree enriched with 41 new features, with full access to metadata. In cases where only part of the audio was analyzed due to overlap, these instances are marked with the "Overlap" feature, and the number of annotations is accordingly reduced. The complete list of the new features [5] added to the corpus is provided in the appendix.[6]

In GREW-MATCH, numerous queries are possible. An example[7] in the syntax-prosody domain shows that when a token is the last element of a prenucleus (`X.Prenucleus=Last`), it tends to exhibit strong final prominence (`X.ProminenceFinal=Strong`) in 65.10% of cases, as shown in Figure 5. In contrast, when the token is the first element of a prenucleus (`X.Prenucleus=Begin`), strong prominence occurs in only 23.20% of cases.[8]

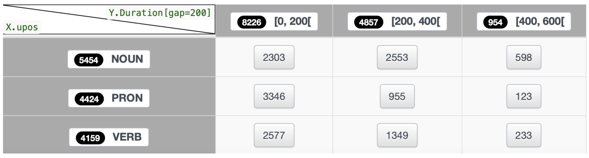


Whether_1 X.ProminenceFinal	1291 0	880 Strong	483 Weak
2236 No	1207	589	411
449 Yes	84	291	72

Figure 5: Results of the query on the position of the IPE and final prominence.

GREW-MATCH also allows querying the newly extracted features, which contain numerical values. For example, this query [9] aims to investigate the correlation between part of speech (`X-[Syl=*]->Y; X[upos=VERB|NOUN|PRON`) and the duration of the final syllable (`Y.duration`). (see Figure 6).

We observed that the final syllables of pronouns are the shortest in 75.63% of the cases, followed by verbs at 61.96%, and nouns at 42.23%. In other words, the final syllables of nouns tend to be longer compared to those of verbs and pronouns.

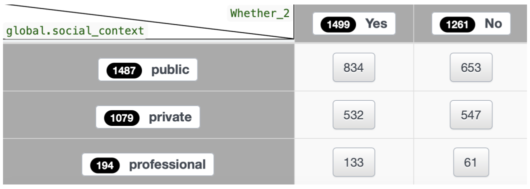


X.upos Y.Duration[gap=200]	8226 [0, 200[	4857 [200, 400[	954 [400, 600[
5454 NOUN	2303	2553	598
4424 PRON	3346	955	123
4159 VERB	2577	1349	233

Figure 6: Results of the query on the verb's position within the IPE and its syllable's duration.

This enriched format facilitates its use in various tasks related to the syntax-prosody interface, and also opens possibilities for sociolinguistic research, even though the original version of Rhapsodie was not initially designed for this purpose.

For instance, the results of a query[10] designed to determine the percentage of IPE boundaries (`X.Period=Last`) that coincide with the end of an illocutionary unit (`X.IU=Last`), according to the social context of the sample, show that the professional social context exhibits the highest alignment between the end of an IU and the end of an IPE (68.56%), followed by the public context at 56.09%. The private context displays the lowest alignment, at 49.30%. (See Figure 7)



global.social_context Whether_2	1499 Yes	1261 No
1487 public	834	653
1079 private	532	547
194 professional	133	61

Figure 7: Results of the query that combines the end of an IPE, an IU and the social context.

5 Conclusion

In this paper, we discussed the process involved in updating the Rhapsodie corpus, which now forms the most feature-rich intonosyntactic treebank available (with the treebank of Naija, (Strickland et al., 2024)). It includes manual, automatic, and semi-automatic annotations, a rare achievement in current research. In the future, we believe this corpus could extend beyond the study of the syntax-prosody interface. Considering that the audio files are available, along with the information provided in the treebank, tasks such as speech modeling or classification in the prosody-syntax-sociolinguistics interface could be explored.

[4] https://universal.grew.fr/?corpus=SUD_French-Rhapsodie-prosody

[5] All information used to describe the Rahpsodie's annotations is drawn from Lacheret-Dujour et al. (2019b) and Bawden and Wang (2015).

[6] Items 1 to 26 were extracted from the original Rhapsodie version; items 27 to 41 were obtained using the tools developed for the Naija intonosyntactic treebank.

[7] universal.grew.fr/?custom=684767a3d2be8

[8] universal.grew.fr/?custom=684ab7e86ff10

[9] universal.grew.fr/?custom=68504ba9ed6b6

[10] universal.grew.fr/?custom=6847617636d58

References

Mathieu Avanzi, Guri Bordal, Anne Lacheret-Dujour, Nicolas Obin, and Julie Sauvage-Vincent. 2019. Chapter 9: The annotation of syllabic prominences and disfluencies. In Anne Lacheret-Dujour, Sylvain Kahane, and Paola Pietrandrea, editors, *Rhapsodie: A Prosodic and Syntactic Treebank for Spoken French*, volume 89 of *Studies in Corpus Linguistics*, pages 158–173. John Benjamins Publishing Company.

Mathieu Avanzi, Anne Lacheret-Dujour, and Bernard Victorri. 2008. Analor. a tool for semi-automatic annotation of french prosodic structure. In *ANALOR. A tool for semi-automatic annotation of French prosodic structure*, pages 119–122.

Rachel Bawden and Ilaine Wang. 2015. *Description of the Rhapsodie TreeBank's Tabular Format: Version: morpho-syntax, micro-syntax, macro-syntax, prosody*. Creation of the tabular format: Rachel Bawden, Ilaine Wang, with the collaboration of Julie Belião. Coordination: Kim Gerdes, Sylvain Kahane. Annotation Platform (Arborator): Kim Gerdes. Microsyntactic annotation: Rachel Bawden, Christophe Benzitoun, Marie-Amélie Botalla, Adèle Désoyer, Sylvain Kahane, Paola Pietrandrea. Macro-syntactic annotation: Christophe Benzitoun, Jeanne-Marie Debaisieux, José Deulofeu, Anne Dister, Florence Lefeuvre, Paola Pietrandrea, Nathalie Rossi-Gensane, Frédéric Sabio, Noalig Tanguy, Bernard Victorri. Prosody: Mathieu Avanzi, Julie Belião, Jean-Philippe Goldman, Anne Lacheret-Dujour, Philippe Martin, Nicolas Obin, Arthur Truong, Bernard Victorri.

Kim Gerdes and Sylvain Kahane. 2017. Trois schémas d'annotation syntaxique en dépendance pour un même corpus de français oral : le cas de la macrosyntaxe. In *Actes de l'atelier « ACor4French – Les corpus annotés du français » (ACor4French 2017)*, Paris, France. ACor4French. LPP, Université Paris 3 Sorbonne Nouvelle & CNRS; Modyco, Université Paris Nanterre & CNRS.

Sylvain Kahane, Kim Gerdes, and Rachel Bawden. 2019. Chapter 4: Microsyntactic annotation. In Anne Lacheret-Dujour, Sylvain Kahane, and Paola Pietrandrea, editors, *Rhapsodie: A Prosodic and Syntactic Treebank for Spoken French*, volume 89 of *Studies in Corpus Linguistics*, pages 49–68. John Benjamins Publishing Company.

Anne Lacheret-Dujour. 2019. Chapter 8: Prosodic annotation of the rhapsodie corpus: Expectations and issues. In Anne Lacheret-Dujour, Sylvain Kahane, and Paola Pietrandrea, editors, *Rhapsodie: A Prosodic and Syntactic Treebank for Spoken French*, volume 89 of *Studies in Corpus Linguistics*, pages 147–155. John Benjamins Publishing Company.

Anne Lacheret-Dujour, Guri Bordal, and Arthur Truong. 2019a. Chapter 11: Derivation of the prosodic structure. In Anne Lacheret-Dujour, Sylvain Kahane, and Paola Pietrandrea, editors, *Rhapsodie: A Prosodic and Syntactic Treebank for Spoken French*, volume 89 of *Studies in Corpus Linguistics*, pages 213–231. John Benjamins Publishing Company.

Anne Lacheret-Dujour, Sylvain Kahane, and Paola Pietrandrea, editors. 2019b. *Rhapsodie: A Prosodic and Syntactic Treebank for Spoken French*, volume 89 of *Studies in Corpus Linguistics*. John Benjamins Publishing Company, Amsterdam / Philadelphia.

Anne Lacheret-Dujour and Bernard Victorri. 2019. Chapter 10: Segmentation into intonational periods. In Anne Lacheret-Dujour, Sylvain Kahane, and Paola Pietrandrea, editors, *Rhapsodie: A Prosodic and Syntactic Treebank for Spoken French*, volume 89 of *Studies in Corpus Linguistics*, pages 175–211. John Benjamins Publishing Company.

Emmett Strickland, Marc Evrard, and Anne Lacheret-Dujour. 2023. SLAM 3: An Updated Stylization Model for Speech Melody. In *Proceedings of the 20th International Congress of Phonetic Sciences (ICPhS2023)*, Prague, Czech Republic. Submitted on 26 Jul 2023.

Emmett Strickland, Anne Lacheret-Dujour, Sylvain Kahane, Marc Evrard, Perrine Quennehen, Bernard Caron, Francis Egbokhare, and Bruno Guillaume. 2024. New methods for exploring intonosyntax: Introducing an intonosyntactic treebank for Nigerian Pidgin. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12207–12216, Torino, Italia. ELRA and ICCL.

A Full version of the new annotations of the intonosyntactic Rhapsodie treebank

1. Layer: the token's inclusion within a layer and its position. Groups of elements that pile up in another element and have the same syntactic position are considered as lists, and the elements inside a list are considered as layers.

{il y a | il y a | il y a | il y a | il y a } *des bons établissements //=*

{'there are | there are | there are | there are | there are } *good schools //=*

`[Rhap_D0002]`

2. Type_para: indicates the kind of relationship that links one layer in a list to another. For instance, para_disfl, para_coord, para_intens, para_dform, para_reform, para_hyper, para_negot.

je travaille à la préfecture de Paris qui { n'est pas connue | ***ˆmais néanmoins existe*** *} "euh" //*

'I'm working at the prefecture of Paris that { isn't well known | ***ˆbut yet exists }*** *"uh" //'*

`[Rhap_D0001, para_coord example]`

3. Type_inherited: in an asymmetrical analysis of lists, there's a dependence with the context in which the lists appears and and inherited dependency that is passed down to the other layers in the list, except in the case of coordination (para_coord), which is exempt from this inheritance.

4. IU: the place of a token inside an illocutionary unit, in BILOU format without the O (Begin, Inside, Last, Unique)

nous < dans le quartier <+ on n'a on n'a pas de lycée > déjà //

'we < in the neighborhood <+ we don't we don't have any high schools > first //'

[Rhap_D0004]

5. Nucleus: the token's inclusion within a nucleus and its place, in BILOU format. Each IU has a nucleus, which is the autonomous constituent that carries the illocutionary force of the IU.

*nous < dans le quartier <+ **on n'a on n'a pas de lycée** > déjà //*

*'we < in the neighborhood <+ **we don't we don't have any high schools** > first //'*

[Rhap_D0004]

6. Prenucleus: the token's inclusion within a prenucleus and its place, in BILOU format.

***moi** < j'ai eu aucun problème scolaire pour mes enfants //*

***'me** < I don't have any school problem for my children //'*

[Rhap_D0002]

7. Innucleus: the token's inclusion within an innucleus and its place, in BILOU format.

*vos journaux (**Jean-Christophe**) qui soulignent également la faiblesse de la mobilisation des électeurs >+ hier //*

*'your newspapers (**Jean-Christophe**) which also underline the poor voter turnout >+ yesterday //*

[Rhap_D2013]

8. Postnucleus: the token's inclusion within a postnucleus and its place, in BILOU format.

*ça a duré dix ans > **le silence autour de moi //***

*'it lasted ten years > **the silence around me//**'*

[Rhap_D2010]

9.IU_parenthesis: the token's inclusion within a in parenthetical IU and its place, in BILOU format.

*il y a une petite rue (+ **^mais dont je ne sais pas le nom //**) une petite rue en & qui tourne un peu //*

*there is a small street (+ **but I don't know its name //**) a little street in & which winds a little //*

[Rhap_M0011]

10. IU_graft: the token's inclusion within a graft, in BILOU format. It occurs when an item appears in an unexpected position.

*vous t~ vous suivez la ligne du tram qui passe vers la & [je crois que c'est une ancienne caserne **"je crois"** //] //*

*'you t~ you follow the tram line which goes toward the & [I think it used to be barracks **"I think"** //] //*

[Rhap_M0003]

11. IU_embedded: the token's inclusion within an embedded unit that takes a governed place in another IU, in BILOU format.

*Marcel Achard écrivait (**[elle est très jolie //= elle est même belle //= elle est élégante //]**)*

*'Marcel Achard wrote (**[she is very pretty //= she is even beautiful //= she is elegant //]**)*

[Rhap_D2001]

12. AssociatedNucleus: the token's inclusion within an associated nucleus and its place, in BILOU format. Associated nucleui are presented as GUs that share distributional properties with nuclei but carry a weak illocutionary force.

*ça < c'est le problème de Paris **"je pense"** //*

*'that < that's the problem of Paris **"I think"** //*

[Rhap_D0004]

13. Intro_IU: The token's inclusion within an IU opener, which is an element distinct from prenuclei, is always at the beginning of an IU and is not microsyntactically dependent on another word.

***et** tu arrives à la fontaine "euh" place Notre Dame //*

***'and** you arrive at the fountain "erm" in Notre Dame square //*

[Rhap_M0001]

14. Period: the token's inclusion within an intonative period and its place, in BILOU format, without the O. Segmentation into IPEs is based on perceptual and acoustic cues. In *tu prends le boulevard euh là qui part de Nef Chavant là le boulevard qui passe à côté d'Habitat* 'you take the boulevard um there that runs from Nef Chavant the boulevard that runs past Habitat' [RhapM0001], *tu* 'you' is annotated as `Period=Begin` and *Habitat* 'Habitat' is annotated as `Period=Last`.

14. Period_tone: the contour of IPE that contains the token. The contour is considered as the first point and last point of a unit, in both points the height of the F0 in relation to the speaker average pith is labeled with five possible levels : very low (L), low (l), middle (m), high (h) and very high (H).

15. Prominence_initial: the degree to which the first syllable of a token is perceived as more salient compared to its surrounding context. This degree is annotated as Weak, Strong, or 0 if the syllable is not prominent.

16. Prominence_final: the degree to which the final syllable of a token is perceived as more salient compared to its surrounding context. This degree is annotated as Weak, Strong, or 0 if the syllable is not prominent.

17. Hesitation: particles such as 'euh' or hesitant syllables, which may exhibit extra-lengthening, infra-low register, or creaky voice.

18. Foot: token's position in the metrical foot, in BILOU format without the O. Within an IPE, every time there is a non-disfluent but prominent syllable, the end of a metrical foot is marked.

19. FootTone: the contour of the last metrical foot of the token.

20. FootType: the category of the token's final metrical foot. The following annotations can be found: dis-strong, dis-weak, filled-dis, filled-pause, silent-pause, strong, tail, or weak.

21. Group: token's position inside a rhythmic group, in BILOU format without the O. A rhythmic group boundary is marked when a RHF (Right Head of the Foot) coincides with the final syllable of a token.

22. GroupTone: the contour of the group rhythmic that contains the token.

23. GroupType: the category of the rhythmic group that contains the token. It can take the same labels as the foot type annotation.

24. Package: token's position within an intonative package, in BILOU format without the O. When rhythmic groups occur in succession, a package is marked by the first group that carries a strong prominence.

25. PackageType: the category of the intonative package that contains the token. Possible annotations include: filled-dis, filled-pause, included, lone, lone-dis-strong, motherless, motherless-dis-weak, silent-pause, or tail.

26. NextBreakLength: the duration of the pause following the token.

27. AvgAmplitude: the mean amplitude of the syllable in decibels.

28. AvgHeightGlo: rough categorical average of the Glo pitch values, with possible values being L, M, and H.

29. AvgHeightLoc: rough categorical average of the Loc pitch values, with possible values being L, M, and H.

30. Duration: syllable's length in milliseconds.

31. Glo: the SLAM3 contour of a global unit, in this case, the token.

31. Loc: the SLAM3 contour of the immediate context of the target unit, in this case, the syllable.

32. MaxAmplitude: the maximum amplitude detected within the syllable in decibels.

33. MeanF0: syllable's mean F0.

34. MeanF0Step: the lowest F0 measurement which would be noticeably higher than the MeanF0, set to two semitones. This is useful for distinguishing between perceptively meaninful pitch differences in continuous data.

35. PitchRangeGlo: A categorical measurement of the pitch difference between the start and end of the `Glo` SLAM contour, with possible values being L, M, and H.

36. PitchRangeLoc: A categorical measurement of the pitch difference between the start and end of the `Loc` SLAM contour, with possible values being L, M, and H.

37. SemitonesFromUtteranceMean: number of semitones between MeanF0 and UtteranceMeanF0.

38. Slope: The slope derived from performing a linear regression

39. SlopeGlo: the slope derived from the `Glo` SLAM value, with possible values including *Rise, Fall, and Flat*.

40. SlopeLoc: the slope derived from the `Loc` SLAM value.

41. UtteranceMeanF0: the utterance's mean F0, annotated in the governing token.

How to Create Treebanks without Human Annotators – An Indigenous Language Grammar Checker for Treebank Construction

Linda Wiechetek **Flammie A Pirinen** **Maja Lisa Kappfjell**
UiT—Norgga árktalaš universitehta
Tromsø, Norway
first.last@uit.no first.last@uit.no first.last@uit.no

Abstract

Creating treebanks for low resource languages is an important task. However, low resource Indigenous language contexts have not only limited resources in terms of text data, but also limited human resources that are available for linguistic annotation. We suggest a work-around by applying a Constraint Grammar operated rule-based dependency parser to do the work of creating a marked-up treebank. However, due to a lot of noise, meaning spelling and grammatical errors in South Sámi written texts, this tool often fails to create complete and correct trees. As a fix to this, we created a grammar checking tool for the most common South Sámi grammatical error types, which improves the quality of the dependency parser significantly. As both literacy and normative standards for most Indigenous languages are much more recent than for majority languages, spelling and grammatical variation and errors are a common source of noise, and the application of a correction tool like ours can be useful in the construction of treebanks for these languages.

1 Introduction

In an extremely low resource language context, treebanks are an important link to developing high level tools that other languages consider standard. Machine-learning based language technology can utilise the treebanks for training and testing new models, and rule-based systems can use them as a gold standard to strive for. In addition they can be used for language comparative tasks, evaluation, etc. Low resource languages like South Sámi, however, are not only low resource in terms of data (< 2 million words) but also lack human resources, which makes manual linguistic annotation of big text corpora impossible. For creating a South Sámi treebank, we therefore applied a Constraint Grammar based dependency annotation tool that can annotate unlimited amounts of text automatically using existing morphological and syntactic tools as their basis. When dealing with low resource Indigenous languages we need to keep in mind that language standards are often still in the process of being developed, and language contact with the majority language influences the way people use their language. South Sámi texts contain a lot of noise in each sentence in terms of typos and non-standard forms, code-switching and sentence structures that resemble literal translations from the majority language rather than using authentic South Sámi syntax. This type of noise is not comparable to the noise in a majority language corpus. It rather reflects the relatively large amount of L2 writers (second language users) in the South Sámi text corpus. As we want a treebank that can also be used for teaching purposes, we would like it to represent mostly L1 language.

Some of these errors and non-standard forms disrupt the sentential dependency structures and prevent our tool from working properly. Especially noun phrase internal errors, case errors and agreement errors lead to broken dependency trees. We therefore suggest the usage of a spelling and *grammar error correction* (GEC) tool as part of the pipeline to create a treebank. All our tools are part of a multi-lingual language resource platform (*GiellaLT*) which provides a common infrastructure for over 150 languages, most of them low-resource and/or Indigenous languages.[1] We manually marked-up error corpora, which we used to identify relevant and frequent errors and created a grammar checking tool that corrects these morpho-syntactic structures. The corrected sentences are then fed into the dependency tool, which create our treebank for South Sámi. South Sámi is an Indigenous language with about 500 speakers, and about 10 percent of these writes the language. This work has been made within a language technology group that started as an initiative of the Sámi Parliament

[1] https://giellalt.github.io and https://giellalt.github.io/LanguageModels.html

20 years ago, which is why we combine both native language and engineering competence. Our main goal is to develop tools for and together with the language community, especially those that are needed in administration and education. This is self-determination in practice, which is also central principle in Sámi endeavors. South Sámi is a Uralic language with interesting syntactic features, such as copula drop, which leaves many sentences without a finite verb, an interesting matter for dependency parsing.

This work is a contribution to creating both proofing tools and a treebank for further research and tool creation. South Sámi did not previously have an annotated treebank, thus our contribution in this work is also that of a new treebank. Our goal was to create in the most efficient way given limited resources, also making sure that language presented therein is authentic but error free. The treebank follows the written standard that is backed by the South Sámi standardization body *Gïelegaaltije* creating a valuable annotated corpus resource. We will in the following present the grammar checking tool, and show how it is integrated into automatic treebank construction of South Sámi.

2 Background

2.1 Language background

South Sámi is an official language in altogether four municipalities in Norway and six municipalities in Sweden. There are approximately 300-600 South Sámi speakers. South Sámi is a morphologically complex language with similar grammatical structures as other Sámi languages. The Sámi languages belong to the Uralic language family, which is unrelated to the Indo-European languages. South Sámi has a number of features that clearly distinguish it from other Sámi languages. South Sámi has even stronger SOV word order than Lule Sámi, and both distinguish between elative and inessive case, which are replaced by locative case in North Sámi. South Sámi typically drops the copula in sentences without pro-drop. It also has nominative plural noun phrases in definite object position, which influences syntactic disambiguation. Negation is more complex than in North and Lule Sámi as South Sámi has a specific paradigm for past tense copula negation verbs that agree with the negation forms. The South Sámi written standard or according to the term of the time, The South Sámi textbook standard, was recommended by the Sámi Language Council in 1976 and was adopted in 1978. (Bergsland and Mattsson Magga, 1993) Some grammatical variants and paradigms have not yet been standardized explicitly by the standardization organ (*Gïelegaaltije*). However, there are written grammars that serve as a basis for teaching and for proofreading. A few grammatical matters are not described in grammars yet, and the grammatical authority lays with the native speaker elders. This knowledge remains to be formalized and presented in a way such that newer speakers that are less exposed to the language can receive the guiding they need to be confident speakers and writers.

Language contact with the Scandinavian majority languages Norwegian and Swedish are further leading to a lot of interference in South Sámi written text. These are clearly marked because they deviate significantly from both Sámi and Uralic morpho-syntax. A clear South Sámi standard is essential for the survival of the language. Without a clear standard new learners lack the confidence to use the language in speech and script and typically chose the safer alternative, the majority language. This means that language planning requires clear choices as regard orthography, lexicon, idioms and grammar to ensure a future for South Sámi and discontinue the colonialization process.

2.2 Technical background

The core pieces of this work are a rule-based dependency analyzer and a grammar checker module. The dependency analyzer is written for the three Sámi languages, North Sámi, Lule Sámi and South Sámi, which is based on a full morphological analysis that is followed by morpho-syntactic disambiguation and syntactic parsing. The syntactic parsing includes only function labels, but no explicit dependencies. Until this step the different Sámi languages have their separate language modules. The dependency structure, however, is added in a common module for all the languages, based on the flat syntactic function tags from the previous module. This work is thoroughly described in Antonsen et al. (2010). The automatic dependency annotation is created bottom up, so that even partial dependency trees can be created if some parts of the sentence contain errors or could not be fully disambiguated. Dependencies build on the same syntactic structure as the grammar checker. They use a specific rule format, which maps dependents to their parents and the other way around based on previously mapped morpho-syntactic labels and

```
SETPARENT:SetObjToRightMv OBJ> TO (*1
(<mv>) BARRIER S-BOUNDARY OR @-FSUBJ>)
;
```

Figure 1: Example rule mapping objects to their right handed verbal mothers

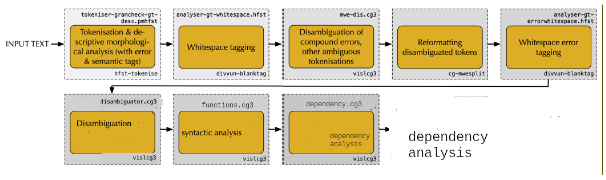

INPUT TEXT
tokeniser-gramcheck-gt-desc.pmhfst
Tokenisation & descriptive morphological analysis (with error & semantic tags)
hfst-tokenise
analyser-gt-whitespace.hfst
Whitespace tagging
divvun-blanktag
mwe-dis.cg3
Disambiguation of compound errors, other ambiguous tokenisations
vislcg3
Reformatting disambiguated tokens
cg-mwesplit
analyser-gt-errorwhitespace.hfst
Whitespace error tagging
divvun-blanktag
disambiguator.cg3
Disambiguation
vislcg3
functions.cg3
syntactic analysis
vislcg3
dependency.cg3
dependency analysis
vislcg3
dependency analysis

Figure 2: Modular structure of the dependency analysis

word order. The parsing of dependencies is based on rules of the type shown in Figure 1, for example where we map the object to a transitive main verb to its right.

The grammar checker module uses the same technology and a similar pipeline. It is specifically written for South Sámi, although some of the error types exist in North and Lule Sámi as well.

Our framework is based on rule-based natural language processing: finite-state morphology (Beesley and Karttunen, 2003) and constraint grammar (Karlsson, 1990). We use the free/ open source VISL CG 3 *constraint grammar* (CG) compiler (Bick and Didriksen, 2015). The linguistic analyses made by the systems include morphological, syntactic and semantic analyses, both on word-level as well as on a dependency graph level. The VISL CG 3 -based dependency analysis has been used in various applications including grammar checking, machine translation, semantic role annotation for various languages like Greenlandic, Danish, Spanish, Portuguese. (Bick, 2019; Rademaker et al., 2017; Bick, 2022)

The VISL CG 3 dependency analysis' foremost goal is not to build a treebank with complete trees, but primarily create another linguistic layer that facilitates the above mentioned tasks when building applications for specific language communities. As trees are created bottom-up, which can leave them partly disconnected, they are not instantly convertible to even better known standards such as Universal Dependencies (UD) (De Marneffe et al., 2021). However, there are previous work that is based on conversion from our annotation system to UD, see for example (Sheyanova and Tyers, 2017; Antonsen et al., 2010) for a North Sámi UD treebank. Automatically generated treebanks need to be verified

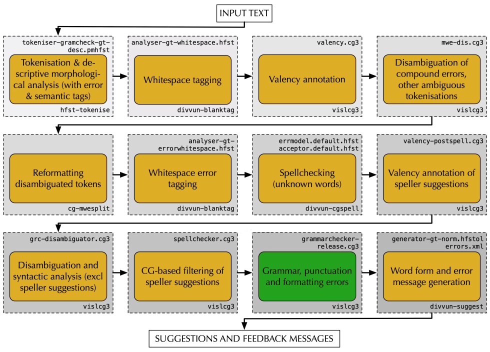

INPUT TEXT
tokeniser-gramcheck-gt-desc.pmhfst
Tokenisation & descriptive morphological analysis (with error & semantic tags)
hfst-tokenise
analyser-gt-whitespace.hfst
Whitespace tagging
divvun-blanktag
valency.cg3
Valency annotation
vislcg3
mwe-dis.cg3
Disambiguation of compound errors, other ambiguous tokenisations
vislcg3
Reformatting disambiguated tokens
cg-mwesplit
analyser-gt-errorwhitespace.hfst
Whitespace error tagging
divvun-blanktag
errmodel.default.hfst
acceptor.default.hfst
Spellchecking (unknown words)
divvun-cgspell
valency-postspell.cg3
Valency annotation of speller suggestions
vislcg3
grc-disambiguator.cg3
Disambiguation and syntactic analysis (excl speller suggestions)
vislcg3
spellchecker.cg3
CG-based filtering of speller suggestions
vislcg3
grammarchecker-release.cg3
Grammar, punctuation and formatting errors
vislcg3
generator-gt-norm.hfstol
errors.xml
Word form and error message generation
divvun-suggest
SUGGESTIONS AND FEEDBACK MESSAGES

Figure 3: Modular structure of the grammar checkers

and fixed by human annotators skilled in the language, this is both by UD guidelines and of course makes a reasonable way to create goldstandards.

The system performing the grammar analysis and correction is built of modules, see Figure 3 for the structure of the grammar checker. The pipeline used for grammatical error corrections includes a syntactical analysis, and the overall system can be used for dependency-based syntactic analysis as well, with slightly different module structure than the one pictured for grammar checking and correction. (Wiechetek and Kappfjell, 2023)

All text data in this work is taken from Sámi international corpus SIKOR (SIKOR, 2025). It contains texts in Sámi languages including South Sámi.

3 A treebank for South Sámi

Our VISL CG 3 dependency analyzer for South Sámi (Wiechetek and Kappfjell, 2023) maps dependencies between word forms that have received a morphological analysis and a syntactical label. Each of these rules builds a partial tree, and combined with each other ideally a full tree is created. However, the tool is also able to construct partial trees, which is useful for atypical sentences, ellipses, headlines, in particular sentences without finite verbs. This is also relevant for South Sámi as copula-drop is a typical feature of the language. (Magga and Mattsson Magga, 2012) It also means that the tool can construct partial trees for sentences that contain spelling and grammatical errors or ommitted words. We ran the dependency parsing tool on 481 sentences and 7,266-token sample corpus to see how many complete trees it is able to construct. 188 of 481 sentences produce complete parse trees. One of these complete trees is displayed in Figure 4 showing the dependency structure of ex. (1). It includes a finite verb and three coordinated infinitives. The vislcg3 output of

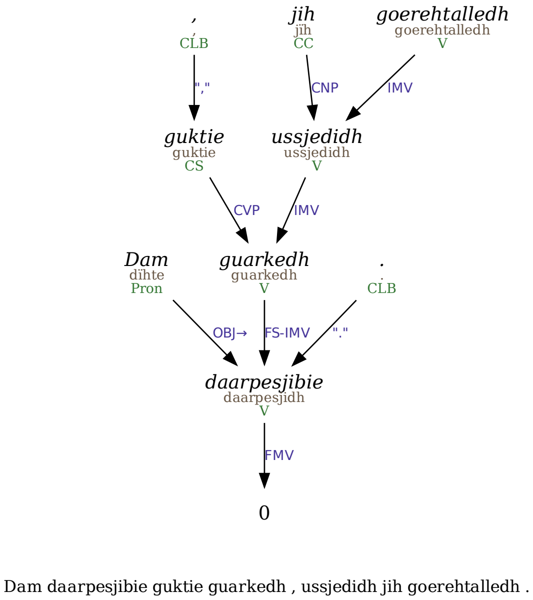


Figure 4: Dependency tree for ex. (1)

```
"<Dam>"
    "dïhte" Pron Pers Sg3 Acc <W:0.0> @OBJ> #1->2
"<daarpesjibie>"
    "daarpesjidh" <mv> V TV Ind Prs Pl1 <W:0.0> @FMV #2->0
"<guktie>"
    "guktie" CS <W:0.0> @CVP #3->4
"<guarkedh>"
    "guarkedh" <mv> V TV Inf <W:0.0> @FS-IMV #4->2
"<,>"
    "," CLB <W:0.0> #5->3
"<ussjedidh>"
    "ussjedidh" <mv> V TV Inf <W:0.0> @IMV #6->4
"<jih>"
    "jïh" CC <W:0.0> @CNP #7->6
"<goerehtalledh>"
    "goerehtalledh" <mv> V TV Inf <W:0.0> @IMV #8->6
"<.>"
    "." CLB <W:0.0> #9->2
```

Figure 5: VISL CG3 dependency output

the dependency analysis is displayed as graphical trees for the purpose of visualization. The original output can be seen in Figure 5, where dependency structures are expressed by absolute numbers after the hashtag for the position of each word pointing to the number of the word they are dependent on. In the case of the finite verb *daarpesjibie* its position in the sentence is *2* and it points to the root *0* (#2->0) It creates a full tree despite the orthographical error in *jih* (should be: *jïh*) as this the morphological analyzer accounts for some of the typical orthographical errors. The object *Dam* should be analyzed as dependent on the infinitive *guarkedh* ‘understand’ instead of *daarpesjibie* ‘need’.

(1) Dam daarpesjibie guktie guarkedh,
that.ACC.SG need.PRS.1.PL for understand,
ussjedidh jih goerehtalledh.
think and investigate
‘We need that to understand, think and investigate.’

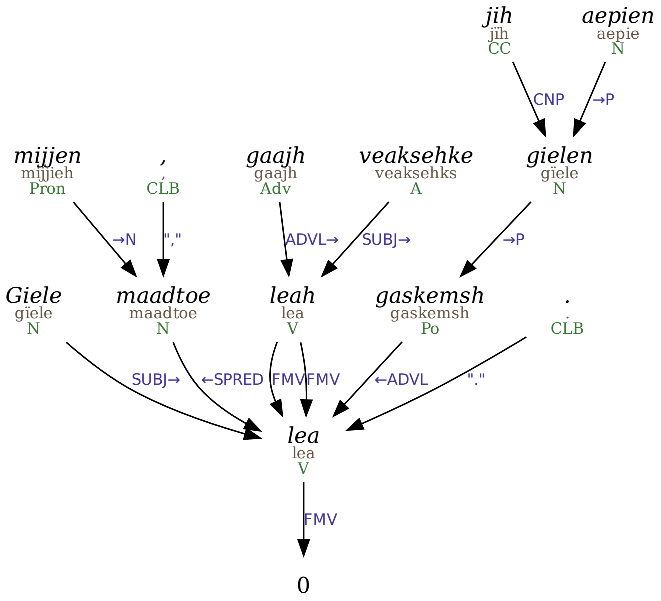
The dependency tree for ex. (2-a) is also complete. However, the dependency structure in Figure 6 shows several errors. The adjective *veaksehke* and the demonstrative pronoun *gaajhkh* should be dependent on the noun *gielen* instead of the finite verb *leah*.

The reason for the partial errors in the dependency structure is one grammatical error in the adjective form *veaksehke* (correct: *veaksehks*) makes it appear a subject in nominative singular instead of an attribute to *gielen*. *Gaajhkh* can therefore not be identified as adverb dependent on the adjective. The morphological analyzer is robust enough to compensate for several spelling errors as the long ‘i’ in three words and misspelled *aepien* (correct: *aerpien*). They still receive a morphological and syntactical analysis.

(2) a. *Giele lea mijjen maadtoe,
language be.PRS.3.SG our foundation,
gaajh veaksehke
incredibly strong.NOM.SG
gielen jih aepien
language.GEN.SG and heritage.GEN.SG
gaskemsh leah.
between be.PRS.3.SG
‘Language is our foundation, there is an incredibly strong connection between language and heritage.’

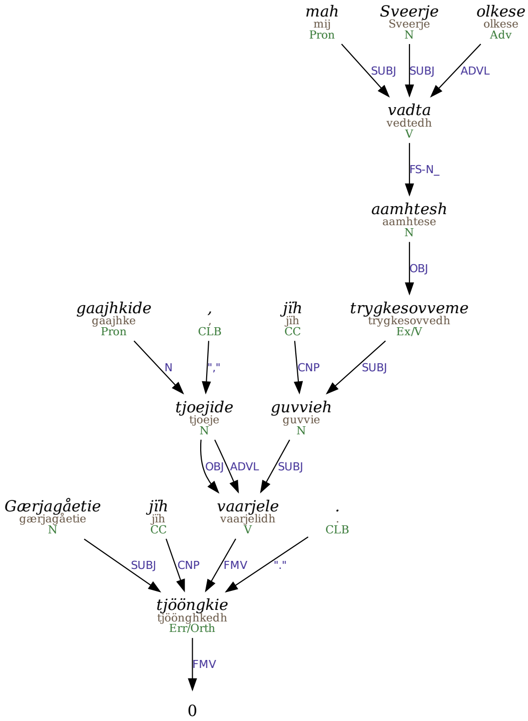
b. Gïele lea mijjen maadtoe, gaajh veaksehks gïelen jïh aerpien gaskemsh leah.

Spelling errors and grammtical non-standard forms are overdimensionally represented in South Sámi written texts. For most majority languages, spelling errors and non-standard forms are filtered out by some kind of proofreading. In addition, writers of majority languages have typically undergone a lot of training and their writing has undergone a lot of proofreading in their respective languages school systems. Figure 7 of a complex sentence including coordinated demonstrative phrases with a relative clause displays a number of these typical errors in South Sámi. Ex. (3-a) shows all errors with their correction in ex. (3-b).



Giele lea mijjen maadtoe , gaajh veaksehke gielen jih aepien gaskemsh leah .

Figure 6: Dependency tree of ex. (2-a)



Gærjagåetie tjööngkie jïh vaarjele gaajhkide tjoejide , guvvieh jïh trygkesovveme aamhtesh mah Sveerje olkese vadta .

Figure 7: Dependency analysis for ex. (3-a)

Morphosntactic errors	334
Syntactic errors	259
Real-word errors	147
Lexical errors	216
Non-word spelling	3,263

Table 1: Error statistics in error annotated text data

(3) a. Gærjagåetie tjööngkie jïh
library collect.PRS.3.SG and
vaarjele gaajhkide
take.care.PRS.3.SG all.ACC.PL
tjoejide, guvvieh jïh
sound.NOM.PL, picture.NOM.PL and
trygkesovveme aamhtesh
printed item.NOM.PL
mah Sveerje olkese
which.NOM.PL Sweden out
vadta.
give.PRS.SG.3
'The library collects and takes care of all sound, images and printed items which Sweden has published'

b. Gærjagåetie tjöööngkie jïh vaarjele gaajhkide tjoejide, guvvide jïh trygkesovveme aamhtesidie mejtie Sveerje bæjhkohte.

The coordinated demonstrative phrase does not have consequent case agreement, the nominative plural nouns *guvvieh* and *aamhtsesh* should be in accusative case just as their coordinated predecessor *tjoejide*. The parsed tree in Figure 7 therefore interprets *guvvieh* as a new subject to *vaarjele* and does not make it a daughter of *tjoejide* as it should be. In addition, the nominative plural relative pronoun *mah* has a case error. It should be accusative *mejtie* in order to be identified as the object of the finite verb *vadta*.

4 Creating a preprocessing tool for dependency structure

In order to create a smoother dependency analysis for South Sámi and facilitate treebank building, we decided to preprocess the text by means of a handwritten spelling and grammar checker for the most common error types. We added a grammatical error annotation layer to SIKOR (SIKOR, 2025). We chose a 182,759-token part of the corpus that had been marked up for spelling errors already, and classified the grammatical error types on top of those. Table 3 shows that the corpus contains altogether 740 errors.

A demonstrative phrase error as explained in ex. (3-a) is marked as a unit. The error is then classified with its morpho-syntactic properties – in this case the nominative plural noun should be in accusative plural – and then the whole phrase is repeated in its corrected form as below.

wrong phrase:

```
gaajhkide tjoejide, guvvieh
jïh trygkesovveme aamhtesh
```

error classification:

```
demphrase,noun,plnom-placc
```

corrected phrase:

```
gaajhkide tjoejide, guvvide
jïh trygkesovveme aamhtesidie
```

Based on our annotation we decided to write rules for the most frequent error types that would potentially affect the dependency analysis of the sentences. Table 2 shows the selected error types with a few of their subtypes. The most common errors after adjective form errors and general case errors (for example in habitive constructions or as a result of valency violations) are typically agreement errors, both between subject and verb and noun phrase internal agreement (including quantifiers and demonstratives).

South Sámi demonstrative phrase and numeral phrases differ from Germanic structures and follow complex rules, which is why errors are common. In demonstrative (and indefinite) phrases typically pronouns and nouns agree in number and case. In numeral phrases, on the other hand, only nominative agrees in number and case. In all other cases, the noun is in singular after all numbers above *one*.

In ex. (4-a), the indefinite pronoun nominative plural *gaajhkh* 'all' needs to be changed to accusative *gaajhkide* ' to all' because of the subsequent accusative noun *maanide* 'children' and its agreement requirements.

(4) a. *Seabradahken dåarjoe
community.SG.INE support
maanasåjhtose edtja
childcare.SG.ILL should.PRS.3.SG
gaajhkh maanide båetedh.
all.PL.NOM child.PL.ILL come.INF
'Community support for childcare should reach all children'

b. Seabradahken dåarjoe maanasåjhtose edtja gaajhkide maanide båetedh.

rule type	**error**	**correction**
demonstrative phrase case agreement	Dem Nom	
numeral phrase agreement	Num N.Nom.Sg.	Num N.Nom.Pl.
numeral phrase agreement	Num N.Pl.	Num N.Sg.
habitive constructions	Nom. copula Nom.	Gen. copula Nom.
infinitive after auxiliary	aux vfin	aux Inf
postposition complement	Acc Po	Gen Po
subject verb agreement	1. Du	3. Pl.
subject verb agreement	3. Pl.	3. Sg.
subject verb agreement	2. Sg.	3. Pl.
subject verb agreement	3. Sg.	3. Pl.
subject verb agreement	Inf.	3. Pl.
phrasal verb lex verb	V Adv	V
unidiomatic phrasal verb	V Adv	V Adv
negation past tense agreement		
negation verb phrase	Neg Inf	Neg Conneg
adjective forms	attr	Nom. Sg.
	attr	Nom. Pl.
	Nom. Sg.	attr
	Nom. Sg.	adv

Table 2: Rule types checked in the South Sámi grammar checking tool

We also need to account for exceptional use of numerals such as in the following sentence (5), where *nulle* 'zero' is actually used as part of a compound 'zero-object' and not as a quantifier.

(5) Voestes aejkien manne **nulle objeekten** bïjre govlim utnim luste goerehtidh maam ij våajnoes aktene raajesisnie.
'The first time I heard about the zero object, I thought it was fun, which wasn't in a sentence.'

Apart from demonstrative phrase, numeral phrase and nominal phrases involving adjectives, also postpositional phrases can alter the dependency structure in parts of the tree. Ex. (6-a) displays a typical case error in dependents of postpositions. In South Sámi, the correct form is genitive case. However, a frequent error is to use accusative case as *dam* 'the' instead of genitive *dan* 'the'. These errors can also involve coordinated noun phrases such as in ex. (7-a).

(6) a. Janne åådtje munnjien dam
Janne get.PRS.3.SG I.ILL that.ACC
bïjre mænngan soptsestidh.
about later talk.INF
Janne can talk to me about it later.

b. Janne åådtje munnjien dan bïjre mænngan soptsestidh.

(7) a. Mijjieh sïjhtebe vuejnedh
we want.PRS.1.PL see.INF
buarastehtemem staaten,
handshaking.ACC state.GEN.SG,
faagesiebrieh jïh barkoevedtijh gaskem
tradeunion.GEN.PL and
juktie destie
employer.GEN.PL between
baalhkajoekehts nyjsenæjjide

'We want to see a handshake between the state, the tradeunion and the employers.'

b. Mijjieh sïjhtebe buarastehtemem vuejnedh staaten, faagesïebri jïh barkoevedtiji gaskem

Other frequent case errors regard habitive constructions such as the one in ex. (8-a), where the possessor role needs to be in genitive case (*Gaajhkesi*) instead of nominative case *Gaajhkesh* 'everyone'. Only then can they be correctly identified as part of the habitive structure in a dependency analysis.

(8) a. Gaajhkesh leah
everyone.NOM.PL are.PRS.3.PL

Dataset	Full trees	Partial
Originals	915	1296
GEC	1390	811
Hand-corrected	1259	948

Table 3: Automatically parsed dependency trees in SIKOR

reaktah årromesæjjan.
right.NOM.PL housing.ILL.SG
'Everbody has the right to a place to live.'

b. Gaajhkesi leah reaktah årromesæjjan.

Verb phrase errors typically regard subject-verb agreement as in examples (9-a) and (10-a), where the verb form needs to be in first person dual instead of first person plural since two and no more people are performing the action. In order to match the verb with its subject, it needs to be in its correct person and number.

(9) a. Daan biejjien Manne jïh Janne
Today I and Janne go.PRS.1.PL
vuelkebe Afrikese,
Africa.ILL.SG vacation.ILL.SG
eejehtæmman

'Today I and Janne are going to Africa for vacation.'

b. Daan biejjien Manne jïh Janne vuelkien Afrikese, eejehtæmman

(10) a. Mænngan Janne jïh manne
Later Janne and I
edtjebe tjaetsieskuvterem
will.PRS.1.PL water.scooter.ACC.SG
vuejedh!
drive.INF
'I and Janne will later drive a water scooter.'

b. Mænngan Janne jïh manne edtjien tjaetsieskuvterem vuejedh!

The following constraint grammar rules in Figure 8 add errortags to (multiple) demonstrative/indefinite pronouns noun combinations and relate them to each other (ADDRELATION) to create a unified error that will be visualized as one error.

```
ADD (&msyn-demphrase-congruence-plnom) TARGET
(Pron Sg Nom) IF (0 Dem OR Indef)(*1 (N Pl Nom)
BARRIER (*) - (Dem Nom) LINK NEGATE 0 (N Sg
Nom));

ADD (&msyn-demphrase-congruence-plnom) TARGET
(Pron Pl Nom) IF (-1 (Pron Dem Pl Nom &msyn-
demphrase-congruence-plnom)) (1 (N Pl Nom));

ADD (&msyn-demphrase-congruence-plnom) TARGET
(N Pl Nom) IF (-1 (Dem &msyn-demphrase-
congruence-plnom) OR (Indef &msyn-demphrase-
congruence-plnom));

ADDRELATION ($2 LEFT) (&msyn-demphrase-
congruence-plnom) TO (-1 (Dem &msyn-demphrase-
congruence-plnom) OR (Indef &msyn-demphrase-
congruence-plnom)) ;
```

Figure 8: Constraint grammar rules adding error tags to demonstrative phrases

```
"<Almetjh>"
	"almetje" N Sem/Hum Pl Nom <W:0.0> @SUBJ> #1->5
:
"<gieh>"
	"gie" Pron Rel Pl Nom <W:0.0> @SUBJ> #2->5
:
"<daesnie>"
	"daesnie" Adv <W:0.0> @ADVL> #3->4
:
"<barkeminie>"
	"barkedh" <mv> V TV Ger <W:0.0> @IMV #4->0
:
"<lea>"
	"lea" <mv> V IV Ind Prs Sg3 <W:0.0> @FMV #5->0
:
"<tryjjes>"
	"tryjjes" A Sg Nom <W:0.0> @<SPRED #6->5
"<.>"
	"." CLB <W:0.0> #7->4
```

Figure 9: Copula drop dependency analysis of ex. (11)

5 Evaluation

We chose a 100 sentence test corpus, part of SIKOR, to manually evaluate the post spell- and grammar checking dependency analysis and got the following results. 73 of 100 sentences received a correct dependency analysis (73%). Of 633 dependencies distributed to word forms – excluding punctuation – 55 human edits were needed to fix the dependencies. This means that 91.3% of the dependencies are correct. 24 of these edits were necessary because the sentence contains copula drop as shown in the dependency analysis of example (11) in Figure 9. Both the non-finite verb *barkeminie* 'working' of the relative clause and the finite verb of the main clause *lea* 'is' go to the root of the sentence, where only the latter should do so.

(11) Almetjh gieh daesnie
people.NOM.PL who.NOM.PL here
barkeminie, lea tryjjes.
working.GER be.PRS.1.SG friendly.NOM.SG
'People who are working here are friendly.'

Copula drop is a known issue in South Sámi describe thoroughly in Ylikoski (2022), and it appears in different forms – the sentence can drop the auxiliary in periphrastic verbal constructions as the one in the previous example, leaving only the non-finite verb form (past participle, gerund etc.). It can also be dropped in copula constructions, leaving only the subject and the predicate. When there are complex sentences with main- and subclause, where the mainclause has copula drop, while the subclause has a finite verb form, the automatic analyzer often analyses the finite verb form of the subclause as the daughter of the root, instead of making it the daughter of the non-finite verbform of the main clause. South Sámi syntax poses challenges to machine-based dependency analysis, which languages with required finite verbs do not, and new solutions need to be carefully investigated.

Other reasons for failing dependencies are remaining spelling and grammar errors (6), and shortcomings in the analysis regarding coordination (7) and finding the correct verbal mother (12).

6 Conclusion

Low resource languages like South Sámi need language resources and treebanks like all other languages. Our approach has taken into account that South Sámi lacks human resources to mark up large amounts of texts to create a treebank by applying a rule-based tool to do so. Instead, we have used our human resources to create and improve rule-based grammar checking and dependency tools so that we can post-edit our treebank with much less effort than creating it from scratch. We have further identified one of the causes of noise in the creation of such resources – spelling and grammatical errors. We therefore enhanced a marked-up error corpus to systematically identify the most frequent grammatical errors that can get into the way of automatic dependency annotation. These include both, errors on the noun phrase and the verb phrase level - demonstrative phrases, numeral phrases, adjectival forms, case errors in habitive constructions and postpositional phrase being a few of them. Based on this analysis we have written rules for all the previous error types to automatically identify and correct these errors and preprocess the input text for the dependency analyzer. We can see that the number of full and partial trees increases with the correction of these grammatical errors, and our current dependency tool gives us 91.3% of correct dependency relations. We were also able to identify the main reasons for remaining flaws in our system. They are related to South Sámi being a copula drop language, which makes it more challenging to identify the roots of these sentences, which can either be a non-finite verb or a nominal phrase. This pecularity of South Sámi will also be interesting when comparing its treebank with the one of other languages. As a next step, we plan to improve our dependency tool and with some human post-editing create the first South Sámi treebank.

We have seen that our method is an efficient way of creating a treebank, a dependency tool and a grammar checker at that same time, all of which can be used as language resources and proofing tools by the South Sámi language community.

References

Lene Antonsen, Trond Trosterud, and Linda Wiechetek. 2010. Reusing grammatical resources for new languages. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Kenneth R Beesley and Lauri Karttunen. 2003. Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*, pages 359–375.

Knut Bergsland and Lajla Mattsson Magga. 1993. *Åarjelsaemien-daaroen baakoegærja*. Iđut, Alta.

Eckhard Bick. 2019. Dependency trees for greenlandic. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, pages 140–148. German Society for Computational Linguistics & Language Technology.

Eckhard Bick. 2022. A modular machine translation pipeline for greenlandic. In *Proceedings of The International Conference on Agglutinative Language Technologies as a Challenge of Natural Language Processing (ALTNLP 2022). CEUR workshop proceedings, Vol 3315. ISSN 1613-0073.*

Eckhard Bick and Tino Didriksen. 2015. Cg-3—beyond classical constraint grammar. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 31–39.

Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational linguistics*, 47(2):255–308.

Fred Karlsson. 1990. Constraint grammar as a framework for parsing running text. In *COLING 1990 Volume 3: Papers presented to the 13th International Conference on Computational Linguistics*.

Ole Henrik Magga and Lajla Mattsson Magga. 2012. *Sørsamisk grammatikk*. Davvi girji, Kárásjohkka.

Alexandre Rademaker, Fabricio Chalub, Livy Real, Cláudia Freitas, Eckhard Bick, and Valeria De Paiva. 2017. Universal dependencies for portuguese. In *Proceedings of the fourth international conference on dependency linguistics (Depling 2017)*, pages 197–206.

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

SIKOR. 2025. SIKOR UiT Norgga Árktalaš universitehta ja Norgga Sámedikki sámi teakstačoakkáldat, veršuvdna 2025-04-23. http://gtweb.uit.no/korp.

Linda Wiechetek and Maja Lisa Kappfjell. 2023. A South Sámi grammar checker for stopping language change. In *Proceedings of the NoDaLiDa 2023 Workshop on Constraint Grammar - Methods, Tools and Applications*, pages 46–54, Tórshavn, Faroe Islands. Association of Computational Linguistics.

Jussi Ylikoski. 2022. South sámi. *The Oxford Guide to the Uralic Languages*, page 113–129.

ComparaTree: A Multi-Level Comparative Treebank Analysis Tool

Luka Terčon
Faculty of Arts, University of Ljubljana / Aškerčeva cesta 2, 1000 Ljubljana
Faculty of Computer and Information Science, University of Ljubljana / Večna pot 113, 1000 Ljubljana
luka.tercon@ff.uni-lj.si

Kaja Dobrovoljc
Faculty of Arts, University of Ljubljana / Aškerčeva cesta 2, 1000 Ljubljana
Jožef Stefan Institute / Jamova cesta 39, 1000 Ljubljana
kaja.dobrovoljc@ff.uni-lj.si

Abstract

ComparaTree is an open-source tool for comparative treebank analysis that combines various methods of quantitative linguistic analysis to provide a general overview of the differences and similarities between two treebanks. The comparison tool covers a range of subfields of linguistic analysis, providing a summary of the differences and similarities in terms of the lexical diversity, n-gram diversity, part-of-speech and dependency relation proportions, syntactic complexity, and syntactic diversity. We explain the various quantitative analyses performed on every level along with the generation of graphical visualizations, which add value by enabling user-friendly comparisons at a glance. We exemplify the comparison process by presenting the results produced by the tool when comparing two treebanks from the Universal Dependencies collection.

1 Introduction

The Universal Dependencies initiative (de Marneffe et al., 2021) has produced a large repertoire of treebanks featuring a consistent, cross-linguistically applicable grammatical annotation format. As of the latest 2.15 release of UD, the collection includes almost 300 treebanks in over 150 languages (Zeman et al., 2024), while at the same time boasting considerable diversity in terms of the various text genres included in the treebanks (Müller-Eberstein et al., 2021), containing also different language modalities such as spoken language (Dobrovoljc, 2022). Given this high degree of diversity, the UD collection is ideal for conducting both intra-linguistic as well as cross-linguistic comparisons, with cross-linguistic studies using UD treebanks becoming especially common (e.g., Nikolaev et al. (2020), Berdicevskis et al. (2018), Levshina et al. (2023)).

Although comparative studies based on Universal Dependencies are becoming increasingly common, there is still a lack of general-purpose tools that facilitate such analyses in a systematic way, as only a handful of specialized tools currently support comparative work. The QuanSyn Python package (Yang and Liu, 2025) supports the analysis of syntactic properties, such as the distribution of parts of speech and dependency relations, within and across treebanks. The STARK tool (Krsnik et al., 2024) enables the extraction of dependency (sub)trees from parsed corpora and supports frequency-based comparisons between datasets. The conllu-diff utility[1] generates statistical summaries of differences between CoNLL-U files, but is limited to individual token-level labels. Beyond these, various processing tools and programming packages are listed on the official UD website,[2] but they are typically not optimized for direct comparative analysis.

While each of the tools mentioned above offers valuable functionality, they are typically limited in scope—focusing on a single linguistic level, requiring programming expertise, or lacking support for user-friendly side-by-side comparison. To address this gap, the present paper introduces ComparaTree, a user-friendly tool for comparative treebank analysis that combines multiple methods of quantitative linguistic analysis. It supports comparisons across lexical diversity, n-gram diversity, part-of-speech and dependency relation distributions, syntactic complexity, and syntactic diversity. ComparaTree also generates visualizations in the form of graphs and diagrams, providing a clear visual overview of the similarities and differences between two treebanks.

In the present paper we first describe the different levels of linguistic analysis for which ComparaTree generates a comparison in Section 2. In Section 3 we exemplify the usage of the tool and

[1] https://pypi.org/project/conlludiff/
[2] https://universaldependencies.org/tools.html

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 129–139
August 28-29, 2025 ©2025 Association for Computational Linguistics

present the format of the results by performing an analysis using two UD treebanks. Finally, in Section 4 we conclude with a discussion of the possible future improvements and extensions.

2 Treebank Comparison

ComparaTree is a tool written in the Python programming language that takes two treebanks in the CoNLL-U format[3] as input and calculates values for various linguistic measures. Although the tool was designed to be used with UD treebanks, in principle any dependency grammar formalism is supported, as long as the treebank used conforms to the standard CoNLL-U format. The source code of the tool is publicly available and can be accessed via a dedicated GitHub repository along with the documentation for its use.[4]

The calculated linguistic measures pertain to five different levels of linguistic analysis: lexical diversity, n-gram diversity, part-of-speech and dependency relation proportions, syntactic complexity, and syntactic diversity. For every level of comparison the tool also outputs a visualization of the results. In the following, we first describe a special process of segment-based averaging in Section 2.1 that is performed for several of the analysis levels. Next we describe the various measures calculated on each level in Section 2.2, while Section 2.3 introduces the various types of resulting visualizations.

2.1 Segment-Based Averaging

For three out of the five analysis levels—lexical diversity, n-gram diversity, and syntactic diversity—a similar methodology is employed to calculate the corresponding measures. The analysis procedure on these three levels involves first splitting the treebank into segments that contain approximately the same number of tokens[5] and subsequently calculating the ratio between the number of unique items and the total number of items in each segment. The final score is obtained by taking the mean of this ratio over all the segments. Each analysis level differs in terms of what is taken as the item for which the ratio is calculated.

This unified analysis method stems from the Type-Token Ratio, a well-established measure of lexical diversity which is obtained by taking the ratio between the number of distinct types (wordforms) and the total number of tokens in a corpus. Although this measure is very commonly used as the default measurement of lexical diversity in many comparative linguistic analyses (e.g., Muñoz-Ortiz et al. (2024) and André et al. (2023)), it is also very sensitive to text length (McCarthy and Jarvis, 2010) and thus might lead to unfair comparisons between treebanks of different sizes. Thus ComparaTree aims to counteract the effect of treebank size using the segment-based averaging technique.

2.2 Levels of Comparison

2.2.1 Basic Comparison

At the most basic level, the tool outputs an overview of the size of both treebanks in terms of the number of tokens contained, the mean sentence length in the number of tokens, and the sentence length standard deviation.

2.2.2 Lexical Diversity

Lexical diversity refers to the amount of variation in the vocabulary used in some corpus and is calculated within ComparaTree using the above-described Type-Token Ratio (henceforth *TTR*). The measure is first calculated for each segment individually and then averaged across all segments. The tool takes the total number of unique lemmas as the number of types in a segment, as this proves more robust when dealing with morphologically richer languages.

2.2.3 N-Gram Diversity

N-gram diversity refers to how prevalent established sequences of words are in a treebank. If a treebank contains fewer unique n-grams (i.e. sequences of *n* consecutive words), this indicates that the corpus is more formulaic and thus has a lower n-gram diversity.

To compute the level of n-gram diversity, the ComparaTree tool first extracts every n-gram[6] in every segment of each treebank along with its corresponding frequency. The tool then calculates the fraction of unique n-grams in the segment, a measure that is also known as the N-gram Diversity score (henceforth *NGD*) (Padmakumar and He, 2024) and averages the score across all segments.

[3] https://universaldependencies.org/format.html

[4] https://github.com/clarinsi/ComparaTree

[5] This segment length value can be adjusted by the user and is set to 1000 tokens by default.

[6] Several values of n can be defined by the user on input to be extracted in a single run of the comparison process.

2.2.4 UD Label Proportions

The tool also calculates the proportional representation for UD part-of-speech and dependency relation labels in each treebank. This involves calculating the ratio between the number of tokens that are assigned a certain label and the total number of tokens in the treebank. We consider the labels which occur more often in one treebank and for which the difference in proportions in both treebanks is the highest the most typical labels for one treebank with respect to the other. In the case of dependency relation labels, dependency subtypes are not counted together with their basic relation types, but are considered as separate categories. ComparaTree also calculates a chi-square test to determine whether the difference between label frequencies in the two treebanks is statistically significant.

2.2.5 Syntactic Complexity

A variety of different measures have been developed which aim to capture the level of syntactic complexity of a text. ComparaTree focuses on the notion of dependency distance as an indicator of syntactic complexity, supporting the calculation of both the Mean Dependency Distance measure (henceforth *MDD*) as well as the Normalized Dependency Distance (henceforth *NDD*) measure. The MDD measures the average distance between syntactically linked words and is a widely-used method that has been the subject of a number of syntactic complexity studies (Ferrer i Cancho, 2004; Futrell et al., 2015). The NDD is based on a similar principle to the MDD, but also takes into account sentence length during calculation and is consequently found to correlate much less with it (Lei and Jockers, 2020; Terčon, 2024). Both measures are calculated on the level of individual sentences and then averaged over the entire treebank.

2.2.6 Syntactic Diversity

The last dimension of analysis provided by ComparaTree is syntactic diversity. It refers to the number of different syntactic patterns that appear in a corpus (De Clercq and Housen, 2017). In the context of treebank comparison, diversity can be represented by the number of different syntactic trees and subtrees that are present. To this end, ComparaTree uses the aforementioned STARK tool for dependency tree extraction (Krsnik et al., 2024) in order to first extract all relevant syntactic trees from each treebank segment. STARK produces a list of all trees and subtrees in a segment along with their associated absolute and relative frequencies based on a number of configuration settings.[7] Once the extraction is complete, ComparaTree uses these lists to calculate a tree diversity score by dividing the number of unique trees in the segment by the total number of trees in the segment. As in the case of lexical diversity and n-gram diversity, the final syntactic diversity score is obtained by taking the mean of the tree diversity scores for all segments.

2.3 Result Visualization

ComparaTree outputs the results both in the form of various lists and tables pertaining to each analysis level, as well as a concise HTML-format summary which consists of two parts: the first is a result summary table containing all the most important measure calculations. For the second part, ComparaTree produces various diagrams in order to visualize the tendencies present in the analyzed data. Examples are given in Appendix A.

In the cases of lexical diversity, n-gram diversity, and syntactic diversity, histograms are generated for both treebanks which show the number of occurrences of each value of the calculated measure—the above-described TTR, NGD, and tree diversity scores—when measured on the level of segments. Similarly, for the basic average sentence length and syntactic complexity—the MDD and NDD scores—histograms are also generated with the values measured on the level of sentences.

In addition, for UD label proportion analysis, the tool generates a barchart showing the proportions for each analyzed UD label with the labels ordered according to the difference in proportion between the two treebanks, placing the labels that are most typical of each treebank at opposite ends of the barchart.

3 Example Comparison: SSJ-UD vs SST-UD

In this section we present an example comparison performed using the ComparaTree tool. The pair of compared treebanks consists of the Slovenian SSJ UD treebank (Dobrovoljc et al., 2017), which represents a balanced sample of written Slovenian, and the Slovenian SST UD treebank (Dobrovoljc and

[7] The STARK package supports various configuration options for tree extraction with the ability to adjust the desired tree size and the type of label that is taken as the tree node. By default, ComparaTree extracts trees of all sizes and considers UPOS tags as tree nodes. These settings can be adjusted by the user via a special configuration input file.

Nivre, 2016), which represents a balanced sample of spoken Slovenian. Both treebanks were provided to ComparaTree as input in the CoNLL-U file format and all the default levels of lingusitic analysis were included. The default segment length of 1000 tokens was used, while for the n-gram analysis only 3-grams were analyzed during this comparison session. In Appendix A, Table 1 presents the result overview table generated by the SSJ vs SST comparison, while Figures 1–8 show the visualizations generated at each individual level of analysis. While a detailed analysis of the results is beyond the scope of this paper, the results plainly illustrate the value of the tool for conducting such multi-level comparisons, as several clear tendencies can immediately be discerned from a single glance at the result summary.

On the **basic level**, Table 1 shows that, while the SSJ treebank contains on average longer sentences than the SST treebank, the sentence length tends to vary much more in SST than SSJ. In terms of **lexical diversity**, the mean TTR score suggests that the spoken language treebank is much less lexically diverse than the written treebank. A similar tendency can be seen in the results of the **n-gram diversity analysis**, where the NGD score for 3-grams is higher in the SSJ treebank compared to SST, indicating that the written treebank has a higher diversity of 3-grams. The differences in **UD label proportions** suggest that nominal phrases are more typical of the SSJ treebank, as nouns, adjectives and adpositions—which commonly occur within nominal phrases—tend to appear more prominently in SSJ. Conversely, particles, interjections, and adverbs—which are commonly connected with non-propositional lexica and other elements that reflect the flow of discourse—appear more typically in the spoken treebank. As for **syntactic complexity**, the values of the MDD and NDD measures exhibit opposite patterns, as the MDD appears to be higher in the SST treebank, while the NDD is higher in the SSJ treebank. Lastly, on the level of **syntacic diversity**, the tree diversity score values show a higher proportion of unique syntactic trees in the written treebank compared to the spoken treebank, suggesting a higher syntactic diversity.

4 Conclusion

In this article we introduced ComparaTree, a tool for comparative linguistic analysis which produces a multi-level comparison of two treebanks. We presented the various levels of linguistic analysis that ComparaTree offers and exemplified its use and output using two treebanks included in the Universal Dependencies treebank collection.

Many functionalities still remain to be added to ComparaTree, which will improve its analysis capabilities. Presently only the UD label proportion analysis is equipped with a statistical significance test that establishes the statistical significance of the observed patterns. In the future, various methods of statistical significance testing along with effect size calculations should be added to other analysis dimensions as well. Although the current presentation of results offers a good glimpse into the tendencies that can be observed in the data, rigorous statistical methods are required to give additional weight to the findings made using ComparaTree.

Additionally, the tool presently only supports pairwise comparisons of two treebanks. Important insights could be gained from comparing more than two treebanks simultaneously, so support for multiple comparisons should be implemented in the future. Such an expansion should also be accompanied by more advanced visualization techniques, which would shed light on different tendencies present in the data and complement the current assortment of histograms and barcharts.

There is also much room to expand the current inventory of measures calculated and range of analyses performed at each level (also in line with new methods that have recently been proposed, such as in Čibej (upcoming)) as well as the potential to expand into other dimensions of linguistic analysis, such as semantics, discourse analysis, etc. Future improvements to the tool in this regard should be dictated by the demand presented by the target users and the broader computational linguistics community.

Acknowledgements

The work described in the present paper was made possible by the Young researchers program, the SPOT project (A Treebank-Driven Approach to the Study of Spoken Slovenian, Z6-4617), the Language Resources and Technologies for Slovene research program (project ID: P6-0411), and the PoVeJMo research project (Adaptive Natural Language Processing with the Help of Large Language Models), all financed by the Slovenian Re-

search and Innovation Agency (ARIS), and the CLARIN.SI research infrastructure.

References

Christopher MJ André, Helene FL Eriksen, Emil J Jakobsen, Luca CB Mingolla, and Nicolai B Thomsen. 2023. Detecting AI authorship: Analyzing descriptive features for AI detection. In *Rome, 7th workshop on natural language for artificial intelligence. NL4AI.*

Aleksandrs Berdicevskis, Çağrı Çöltekin, Katharina Ehret, Kilu von Prince, Daniel Ross, Bill Thompson, Chunxiao Yan, Vera Demberg, Gary Lupyan, Taraka Rama, and Christian Bentz. 2018. Using Universal Dependencies in cross-linguistic complexity research. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 8–17, Brussels, Belgium. Association for Computational Linguistics.

Jaka Čibej. upcoming. A computational method for analyzing syntactic profiles: The case of the ELEXIS-WSD parallel sense-annotated corpus. In *SyntaxFest 2025*, Ljubljana, Slovenia.

Bastien De Clercq and Alex Housen. 2017. A cross-linguistic perspective on syntactic complexity in L2 development: Syntactic elaboration and diversity. *The Modern Language Journal*, 101(2):315–334.

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

Kaja Dobrovoljc. 2022. Spoken language treebanks in Universal Dependencies: an overview. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1798–1806, Marseille, France. European Language Resources Association.

Kaja Dobrovoljc, Tomaž Erjavec, and Simon Krek. 2017. The Universal Dependencies treebank for Slovenian. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, pages 33–38, Valencia, Spain. Association for Computational Linguistics.

Kaja Dobrovoljc and Joakim Nivre. 2016. The Universal Dependencies treebank of spoken Slovenian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1566–1573, Portorož, Slovenia. European Language Resources Association (ELRA).

Ramon Ferrer i Cancho. 2004. Euclidean distance between syntactically linked words. *Phys. Rev. E*, 70:056135.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Luka Krsnik, Kaja Dobrovoljc, and Marko Robnik-Šikonja. 2024. Dependency tree extraction tool STARK 3.0. Slovenian language resource repository CLARIN.SI.

Lei Lei and Matthew L. Jockers. 2020. Normalized dependency distance: Proposing a new measure. *Journal of Quantitative Linguistics*, 27(1):62–79.

Natalia Levshina, Savithry Namboodiripad, Marc Allassonnière-Tang, Mathew Kramer, Luigi Talamo, Annemarie Verkerk, Sasha Wilmoth, Gabriela Garrido Rodriguez, Timothy Michael Gupton, Evan Kidd, Zoey Liu, Chiara Naccarato, Rachel Nordlinger, Anastasia Panova, and Natalia Stoynova. 2023. Why we need a gradient approach to word order. *Linguistics*, 61(4):825–883.

Philip M McCarthy and Scott Jarvis. 2010. MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2021. How universal is genre in Universal Dependencies? In *Proceedings of the 20th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2021)*, pages 69–85, Sofia, Bulgaria. Association for Computational Linguistics.

Alberto Muñoz-Ortiz, Carlos Gómez-Rodríguez, and David Vilares. 2024. Contrasting linguistic patterns in human and LLM-generated news text. *Artificial Intelligence Review*, 57(10):265.

Dmitry Nikolaev, Ofir Arviv, Taelin Karidi, Neta Kenneth, Veronika Mitnik, Lilja Maria Saeboe, and Omri Abend. 2020. Fine-grained analysis of cross-linguistic syntactic divergences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1159–1176, Online. Association for Computational Linguistics.

Vishakh Padmakumar and He He. 2024. Does writing with language models reduce content diversity? *Preprint*, arXiv:2309.05196.

Luka Terčon. 2024. Uporaba šestih mer skladenjske kompleksnosti za primerjavo jezika v govornem in pisnem korpusu. *Jezikovne tehnologije in digitalna humanistika*, pages 668–686.

Mu Yang and Haitao Liu. 2025. QuanSyn: A package for quantitative syntax analysis. *Journal of Quantitative Linguistics*, 0(0):1–18.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Arofat Akhundjanova, Furkan Akkurt, Gabrielė Aleksandravičiūtė, Ika Alfina, Avner Algom, Khalid Alnajjar, Chiara Alzetta, Erik Andersen, Matthew Andrews, and 633 others. 2024. Universal Dependencies 2.15. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

A Example ComparaTree Output

Metric	SSJ	SST
Basic		
Total # of tokens	267,097	98,393
Total # of sentences	13,435	6,108
Average tokens per sentence	19.881	16.109
Standard deviation of tokens per sentence	12.766	17.881
Lexical Diversity		
Average Segmental Type-Token Ratio	0.482	0.297
Segmental Type-Token Ratio standard deviation	0.039	0.050
N-Gram Diversity		
Average Segmental 3-gram Diversity Score	0.995	0.984
Segmental 3-gram Diversity Score standard deviation	0.006	0.010
UD Label Proportions		
Largest part-of-speech tag proportion differences	NOUN – 0.10 ADJ – 0.05 ADP – 0.03 PROPN – 0.02	PUNCT – 0.08 PART – 0.04 INTJ – 0.03 ADV – 0.03
Largest dependency relation proportion differences	nmod – 0.05 amod – 0.05 case – 0.04 obl – 0.02	punct – 0.08 advmod – 0.03 discourse – 0.03 root – 0.01
Syntactic Complexity		
Average Mean Dependency Distance	2.572	2.738
Mean Dependency Distance standard deviation	0.925	1.099
Average Normalized Dependency Distance	1.146	0.850
Normalized Dependency Distance standard deviation	0.509	0.452
Syntactic Diversity		
Average Segmental Tree Diversity Score	0.730	0.689
Segmental Tree Diversity Score standard deviation	0.033	0.052

Table 1: Table showing a summary of every measure calculated at each level of linguistic analysis as provided by ComparaTree for the comparison between the SSJ and SST UD treebanks. The *UD Label Proportions* subdivision presents the four labels for which the proportion difference between the two treebanks is the greatest, thus presenting the labels that are most typical of one treebank with respect to the other. The absolute values of the differences are provided next to the label names.

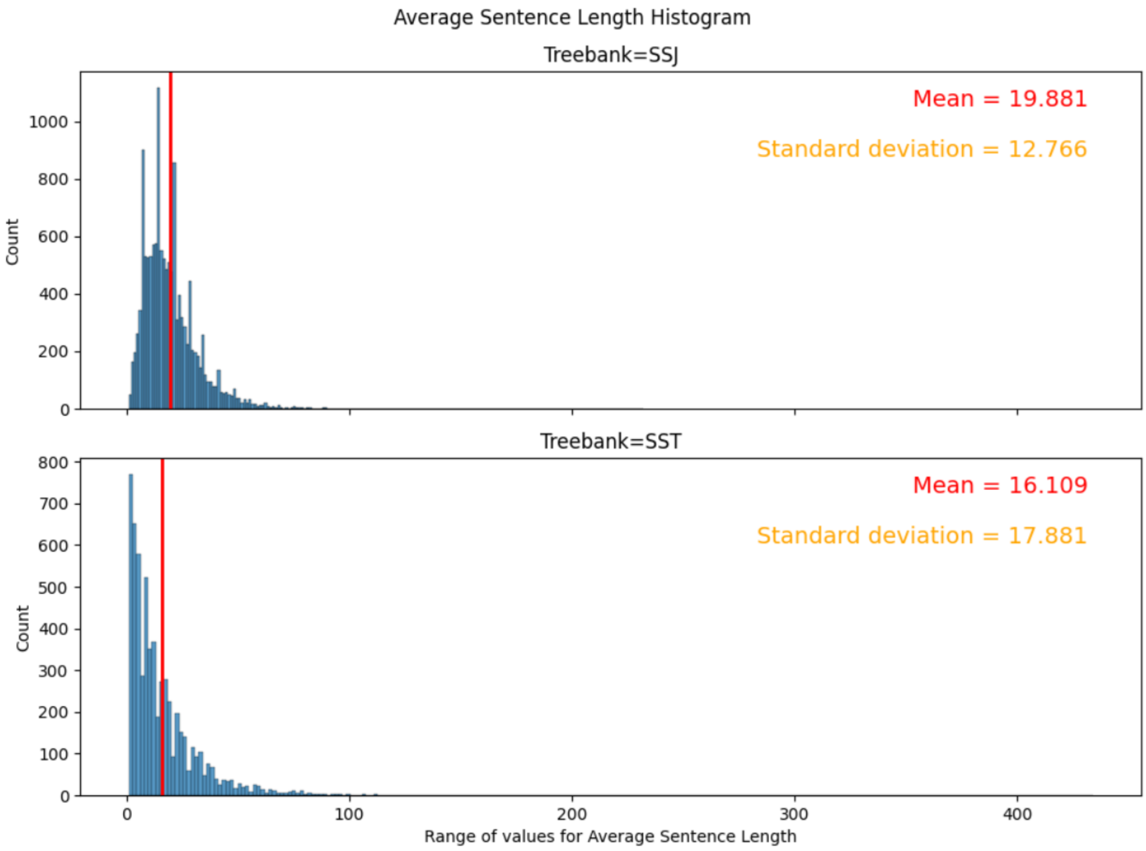


Figure 1: Histogram showing the frequency distribution for the sentence length in the number of tokens for both treebanks. The x axis represents the range of values for the sentence lengths. The blue bars represent the number of observations for each value of the measure. The red vertical line represents the mean sentence length.

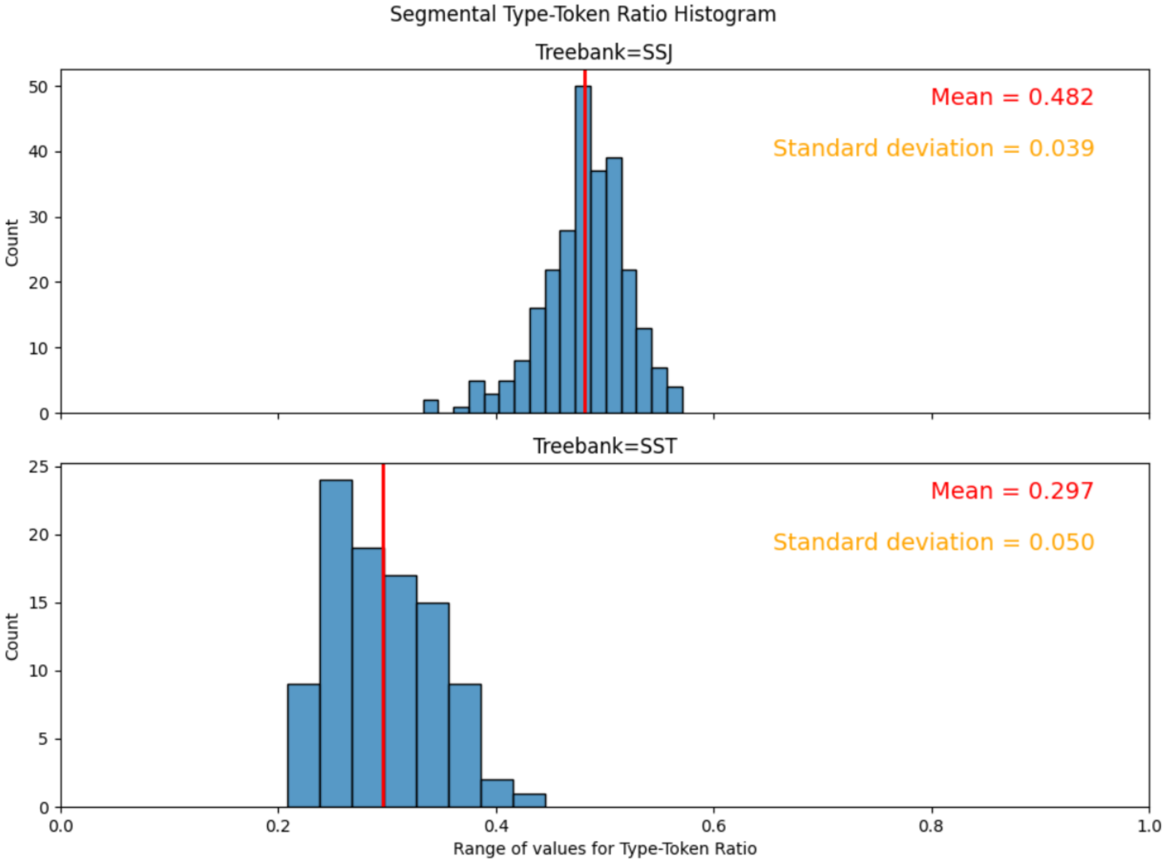


Figure 2: Histogram showing the frequency distribution for the per-segment Type-Token Ratio in both treebanks. The x axis represents the range of values for the Type-token Ratio. The blue bars represent the number of observations for each value of the measure. The red vertical line represents the mean of the Type-Token Ratio over all segments.

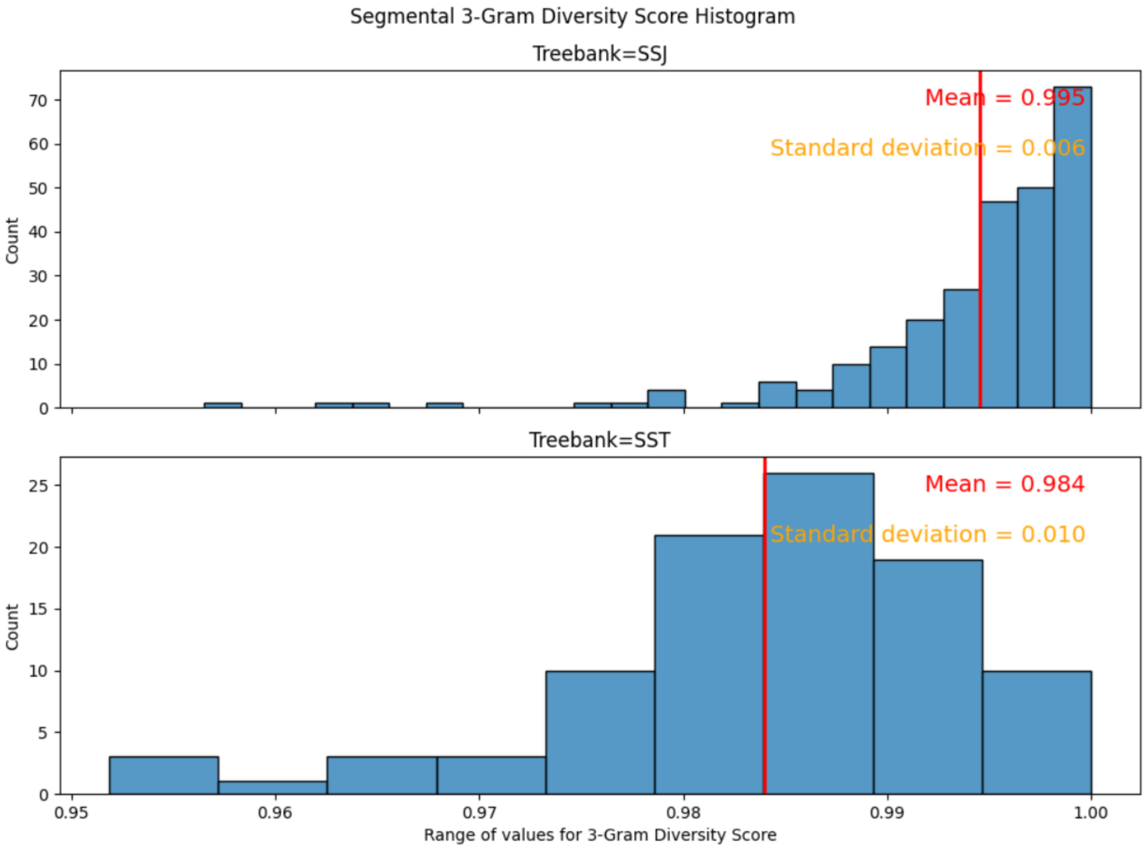


Figure 3: Histogram showing the frequency distribution for the per-segment 3-Gram Diversity Score in both treebanks. The x axis represents the range of values for the 3-Gram Diversity Score. The blue bars represent the number of observations for each value of the measure. The red vertical line represents the mean of the 3-Gram Diversity Score over all segments.

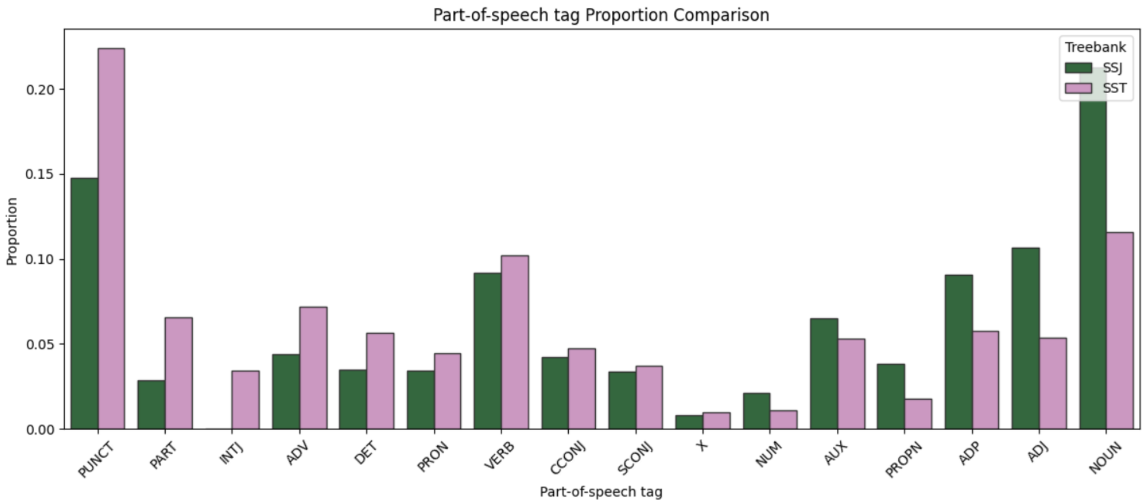


Figure 4: Barchart showing the proportion of every UPOS tag for each treebank. The ordering of the tags is determined by the difference between the tag proportions between the two treebanks, with the tags on the left end being more typical (i.e. occurring with a higher proportion difference) of SST, while the tags on the right end being more typical of SSJ.

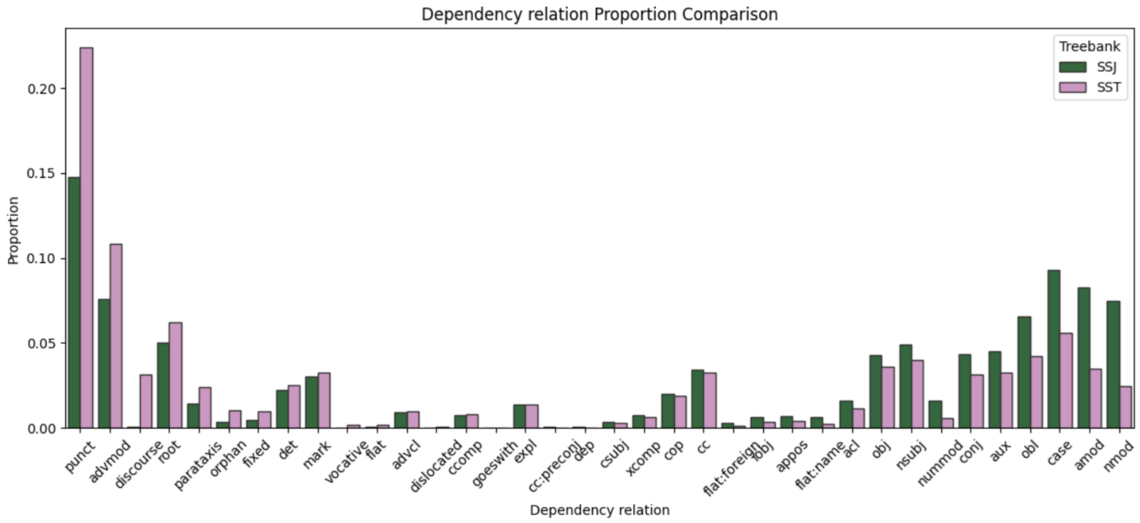


Figure 5: Barchart showing the proportion of every dependency relation tag for each treebank. The ordering of the tags is determined by the difference between the tag proportions between the two treebanks, with the tags on the left end being more typical (i.e. occurring with a higher proportion difference) of SST, while the tags on the right end being more typical of SSJ.

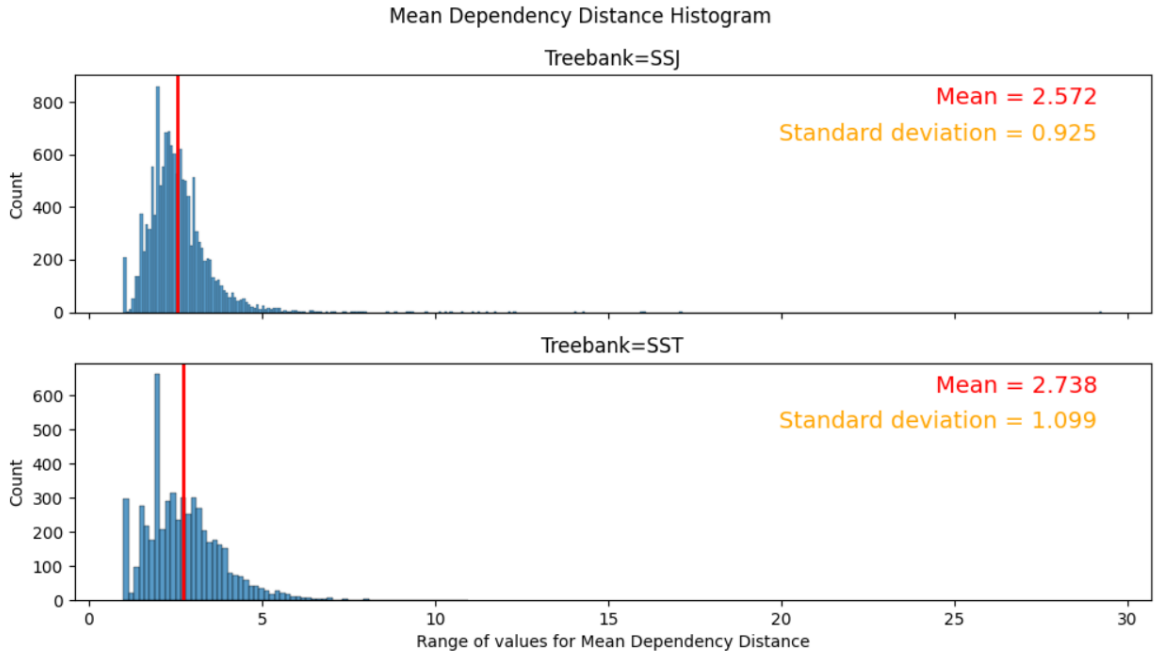


Figure 6: Histogram showing the frequency distribution for the per-sentence Mean Dependency Distance in both treebanks. The x axis represents the range of values for the Mean Dependency Distance. The blue bars represent the number of observations for each value of the measure. The red vertical line represents the mean of the Mean Dependency Distance over all sentences.

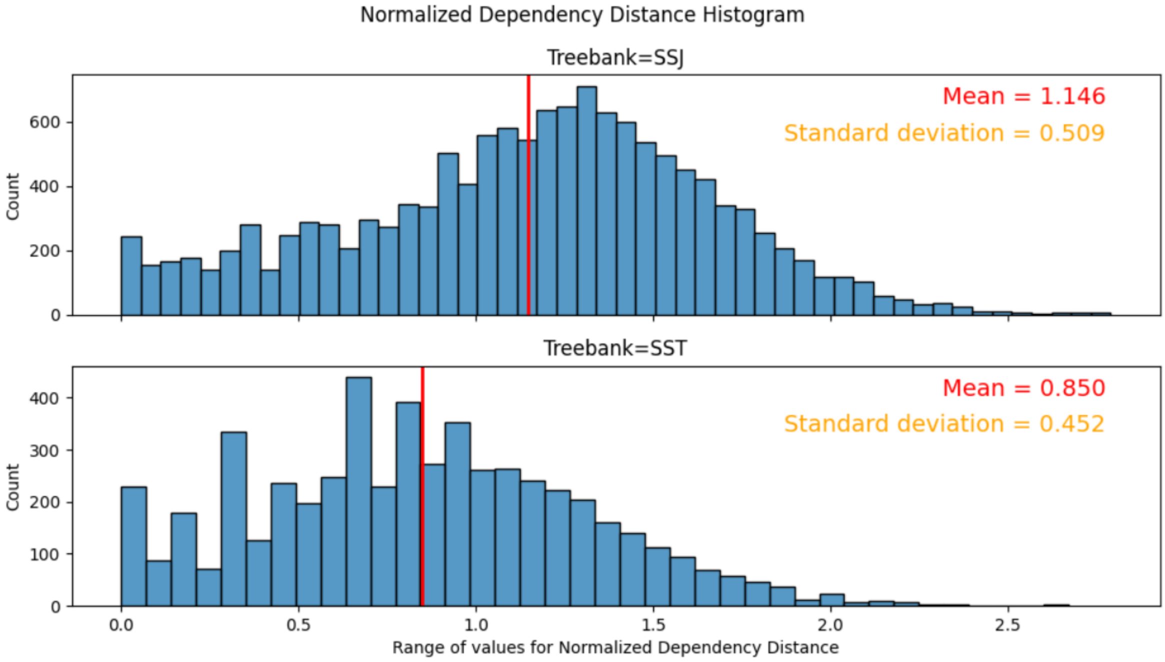


Figure 7: Histogram showing the frequency distribution for the per-sentence Normalized Dependency Distance in both treebanks. The x axis represents the range of values for the Normalized Dependency Distance. The blue bars represent the number of observations for each value of the measure. The red vertical line represents the mean of the Normalized Dependency Distance over all sentences.

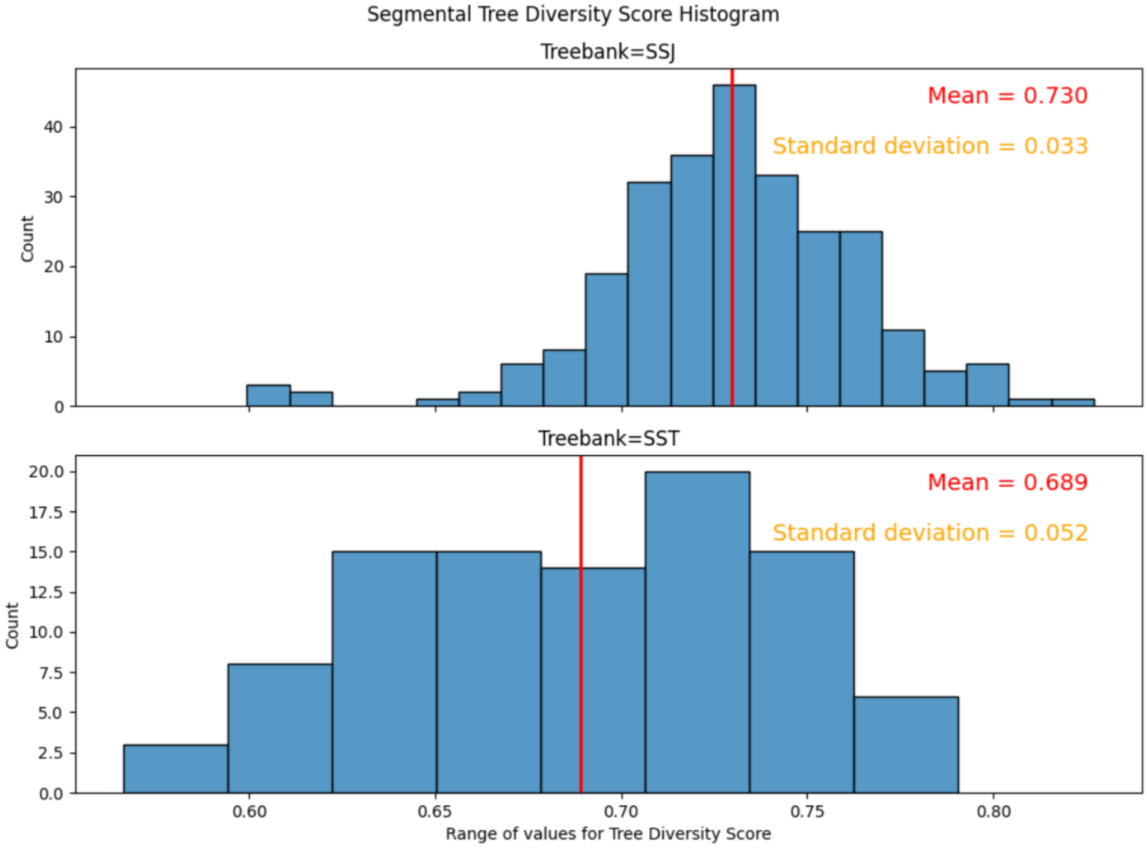


Figure 8: Histogram showing the frequency distribution for the per-segment Tree Diversity Score in both treebanks. The x axis represents the range of values for the Tree Diversity Score. The blue bars represent the number of observations for each value of the measure. The red vertical line represents the mean of the Tree Diversity Score over all segments.

Universal Dependency Treebank for a low-resource Dardic Language: Torwali

Naeem Uddin, Daniel Zeman
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics, Prague
naeemuddinhadi@gmail.com, zeman@ufal.mff.cuni.cz

Abstract

This paper presents and discuss the linguistic phenomena encountered in the development of the ongoing first ever universal dependency treebank for the Torwali Language. Torwali belongs to the Kohistani sub-group of Dardic Indo-Aryan languages, and is considered an endangered (Moseley, 2010) and indigenous language, which makes it extremely low-resourced in terms of linguistic and computational resources. With the aim of including Torwali in Universal Dependencies (UD) (de Marneffe et al. 2021), we are annotating a diverse set of example sentences for POS tags, features and dependency relations.

Keywords: Torwali, Universal Dependencies, treebank, POS tags

1 Introduction

Northern Pakistan is characterized as one of the most linguistically diverse regions with around 30 indigenous language communities (Liljegren and Akhunzada, 2017), among which many are endangered languages. Torwali is one of such communities, located in Swat Kohistan, which belongs to the Kohistani sub-group of the Dardic Indo-Aryan languages (Ullah, 2004). It has two dialects (the Bahrain and Chail dialects), with a total of over 100,000 speakers approximately.

Torwali [ISO 639-3: trw], is a marginalized and low-resource language written in right-to-left Perso-Arabic script. There is a glossonymic variation between Torwalik (Biddulph, 1880), Torwali (Grierson, 1929), Kohistani (Raverty, 1862) and Torwali-Kohistani (Rensch 1992) across time, literature and communities. There are some resources of Torwali language description (Grierson, 1929), a study of linguistic features (Lunsford, 2001) and a structured lexical database (Ullah, 2004). But, when it comes to resources for computational processing of Torwali, there is a lack of robust resources of morphology and grammar.

UD treebank data is useful for downstream tasks in NLP, including semantic parsing (Reddy et al., 2017) and natural language understanding (Schuster and Manning, 2016), and syntactic corpora are useful for linguists studying language typology and change (Levshina, 2019). As there is no coverage in UD to date for Dardic languages, this work upon completion and inclusion in UD, will help in creation of treebanks for other Dardic languages as well, and will help mitigate the bias towards the “big” Indo-European languages (Nivre et al. 2020) in the UD.

This paper discusses the approach to build a Torwali dependency treebank as well as syntactic constructions and grammatical structure of an extremely low-resourced and less-studied language. The treebank is currently under development with a target to cover 500 sentences taken from Inam Ullah’s Torwali-English-Urdu Dictionary**.**

2 Data collection and annotation

The data for this study was extracted from the lexical database of Torwali in Toolbox format created by Inam Ullah (Ullah, 2004), which encodes linguistic information using tagged field markers. Only example sentences were extracted, as they provide naturalistic usage data for lexical items in context. A custom script was developed to parse the Toolbox file and isolate fields containing example sentences, while discarding other lexical metadata. Minor formatting inconsistencies, such as irregular punctuation or spacing, were corrected to ensure uniformity across sentences.

After preprocessing, the extracted sentences were converted into the CoNLL-U format, which consists of a structured 10-column format designed to facilitate manual annotation. This format enables the inclusion of tokenization, morphological features, and syntactic

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 140–147
August 28-29, 2025 ©2025 Association for Computational Linguistics

dependencies. The last column of the CoNLL-U format is used, among other things for transliteration, allowing for a consistent representation of the Torwali text in a Latin-based alphabet for easier analysis and cross-linguistic comparison.

Since we have extracted almost all the sentences from Toolbox file, the current workflow involves a rather large amount of manual work. The treebank has been annotated by a native speaker of Torwali, the first author of this paper.

3 Morphology

Torwali has a complex morphology because it is basically a fusional language which uses several strategies like stem modification, reduplication and existence of words in inflected form, derived form, compound form and root form.

In Torwali, nouns are inflected for number and case and the stem can be joined by an optional plural suffix and an optional oblique case marker. Torwali uses several strategies to mark plurality, but the primary morphological method is tone along with verb agreement like for most of the singular nouns have a tone with rising pitch from low-to-high and their plural counterparts have a tone with low pitch.

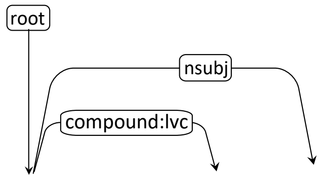
Torwali verbs inflect for tense, aspect, mood and gender and most of the verb forms make gender and number distinction only, no distinction for person. Torwali has three tenses: present, past and future. We did not record any distinction between simple present and present continuous tense in Torwali. It also encodes elevation in motion verbs, distinguishing 'going up' and 'going down' based on real altitude. This reflects the speakers' mountainous environment, with multiple verbs (e.g., **واد**/wad*/came down-went down/*, and **اوگھاد**/ughād*/came up-wen*t *up*) conveying nuanced direction, though all translate as 'go' or 'come' in English.

4 Syntactic features

4.1 Word order and head position

Torwali language exhibits a SOV (Subject-Object-Verb) word order, where the verb consistently appears in clause-final position, so it is a head-final language; in accord with that, it uses pospositions rather prepositions. This structure is further complicated by gender agreement, in which the verb cross-references the gender and number of its preceding subject or object, creating a tight sytactic bond between the verb and its arguments. Consider the following example sentence (1). [1]

(1)



کوییٔ **بات** **جوَوٗی**

koi *bat* *ǰəwəi*

do.PRES.SG.F talk woman

VERB NOUN NOUN

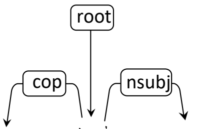
“ The woman talks”

In (1), the gender is marked by attaching /**یئ** /i/ to verb /**کو**/ko/, if the preceding subject is feminine and /**دو**/du/ is attached to verb if the preceding subject is masculine.

4.2 Copula

In Torwali, the copula is morphologically rich and obligatorily present in all major predicate structures, including predicate nominals, locative clauses, and possessive constructions. The copula inflects for gender and number, typically distinguishing masculine singular, feminine singular, and plural but it does not mark future tense, which is expressed through other verbal strategies. We annotated the copula AUX and attached it to the nominal or adjectival predicate via the cop relation as shown in (2), while the predicate itself serves as the head of the clause, and the subject is attached with it.

(2)



تھو ڈاکٹر سے

thu *ḍakṭər* *se*

is.M doctor he.3SG

AUX NOUN PRON

“He is a doctor”

For present tense the copula is **تھو**.M.SG/*thu,* **چھی**.F.SG/*čhi* and **تھی**.PL/*thi,* and for past tense

[1] The words in all tree diagrams are ordered right-to-left in accord with the Torwali original. This has to be borne in mind when reading the English glosses.

اشو.M.SG/*ašu* , أشی.F.SG/*æši* and اشی.PL/*aši*.

4.3 Number

As mentioned in section 3, nouns in Torwali are inflected for number and use a mix strategy which includes tone, joining an optional plural suffix and stem modification. The nouns نار/*nar*/dance.SG and نأر/*nær*/dance.PL demonstrate a singular-plural distinction, which shows modification of stem. Interestingly, نأر/*nær* can also function in contexts when combined with verbs describing continuous or interrupted actions:

(3) گأ لار مے نأر جوَؤی
gæ lār me nær ǰəwəi
go.PAST.F fall in dance woman.SG
" The woman fell while dancing"

This dual usage suggests that نأر may also mark imperfective aspect or dynamic action rather than strict plurality. Now, below is a sentence where نأر marks plurality.

(4) کوؤدد نأر جوَؤی
kowdud nær ǰəwəi
do.PAST dances.PL woman.SG
" The woman used to perform dances"

Furthermore, the use of tone may also be used to mark plurality along with the argument the verb takes. From the example below in (5), to mark plurality, there is a change in tone, usually from high-to-low for vowel ending plurals like جوَؤی/ *ǰəwəi* /women, along with the verb /کو/ ko/ taking an argument /دی /di/ i-e:

(5) کودی نار جوَؤی
kodi nar ǰəwəi
do.PRES.PL dance woman.PL[tone=HL]
" The women dance"

There is no orthographic distinction between these tonal items such as in case of جوَؤی/ *ǰəwəi*, which can be plural or singfular depending on tone. Such constructions can lead to ambiguities while annotating them in CoNLL-U format and can confuse the parser as well, which possibly could be addressed by adding a custom feature like Tone="_" in the `FEATS` column.

4.4 Tonal distinctions and minimal pairs

In Torwali minimal pairs exist via tonal contrasts, with identical forms differing in pitch, following are some of the examples:

ژاد/*ẓàd*/morning/[HL] vs ژاد/*ẓād*/blood/[H]

ماش /*màš* /uncle[HL] vs ماش/*māš*/fish.sg[H]

پھاپ/*phāp*/lung[H] vs پھاپ/*phàp*/uncle[HL]

Agreement patterns further complicate this; there are some variants of lexical items which the native speakers do not treate as separate lexical items such as (e.g., ژاد/ژات/*ẓād*/*ẓāt*^{h}/blood) and there are other such examples as well. But, as of now there is no evidence that such items are regionally, or phonologically conditioned, and this is early to say that such words show dialectal or contextual allomorphy.

Apart from tonal minimal pairs, interesting observation regarding phoneme minimal pairs is the accusative forms of:

تیس /*tes*/him/her/, کیس /*kes*/whom, میس /*mes*/this

and their oblique counterparts,

تِس /*tis*/him/her, کِس/*kis*/who/whom, مِس/ *mis*/this

the only difference between them is the change from /e/ to /i/.

As the tone plays a very significant role in the language, it affects many areas of Torwali grammar, therefore, there is a need to analyze tone in Torwali from different angles (Lunsford, 2001). Furthermore, this tone-driven ambiguity complicates parsing if two words are only differentiated by tone, and that tone is not encoded, parsers can easily confuse them. We have not yet encoded pitch or tone in our annotation.

4.5 Gender

As presented in examples sentences (1) from section 4.1, Gender in Torwali is not marked on nouns but is instead determined by verbs. The verb agrees with the gender of the preceding noun or argument, as seen in (6) and (7).

(6) پودو او سے
pudu u se
drink.PRES.M water 3rd.SG
VERB NOUN PRON
"He drinks water"

(7) پویئ او سے
pui *u* *se*
drink.PRES.F water 3rd.SG
VERB NOUN PRON
"She drinks water"

This alignment suggests a head-marking pattern where verbs encode gender information. In order to annotate such a sentence, the gender of almost all 3rd person pronouns must be determined by verb ending and/or suffix, as shown in (6) and (7), because سے*/se = he/she/it/they*, which can either be masculine, feminine, singular or plural.

Other examples where the adjective agrees with the noun it modifies:

(8) تهو گهن باڈ
thu *ghən* *bad*
is.M large/big.M stone
VERB ADJ NOUN
"Stone is big/large"

(9) چهی گهین نهیت
či *ghen* *nhet*
is.F large/big.F river
VERB ADJ NOUN
"River is big/large"

Although, grammatical gender can be distinguished by biological gender such as
/داد/ *dad/grandfather/* and /دأت /*dæt/ grandmother/.*
It is also noted that there is no gender marker for 3rd person in future tense.

(10) بن-نین کتاب تی
bən-nin kitab ti
read.FUT.M.F book 3SG
VERB NOUN PRON
"He will read the book"

In above example, neither the 3rd person /تی/*ti*/ nor the verb gives any information about the gender of the subject. Verb here has a suffix نین /*-nin*, for future tense.

4.6 Gender and number neutralization

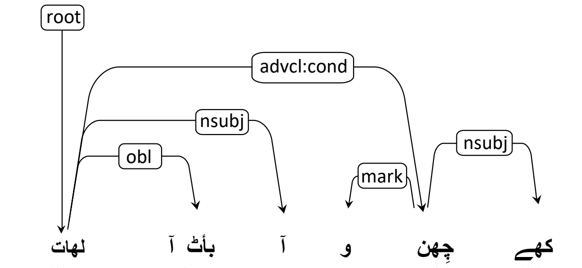
As discussed in sections 4.1, 4.2, gender of nouns and pronouns is marked by verbs and auxiliaries. But the invariant past continuous tense suffix /دود/*dud*/ i-e from لَھنگُودُود/*lhəŋudud*/(was/were entering) shows complete gender and number neutrality. Which contrasts sharply with Urdu's gender and number-sensitive past auxiliaries (/تھا/*tha*/تھی/*thi*/تھے/*the*/, indicating simpler inflectional morphology.
A strange phenomenon arises when the gender and number-invariant third-person pronoun (سے/*se*/ = he/she/it/they) combines with an invariant past continuous tense verb with the suffix /دود/*dud*/ such as لَھنگُودُود /*lhəŋudud*/ ("was/were entering"). This combination makes it particularly difficult to encode explicit grammatical information, like gender and number within the UD framework. While UD can accurately represent this lack of marking, it highlights the degree to which some languages rely on context rather than morphology to convey these fundamental grammatical categories. Below example illustrates the behavior.

(11)
لَھنگُودُود یے دکان سے
lhəŋudud ye dukan se
entering.PST to shop 3SG/3PL.MASC/FEM
"He was entering the shop"

4.7 Conditional constructions

Torwali encodes conditionals morphologically as well as well syntactically. Here is an example of morphological marking of conditional mood with the conditional suffix: -و

(12)



lhat bæṭa a o čhin khe
emptied bundle.Obl.Sg me if.COND break.PRES rope
VERB NOUN PRON SCONJ VERB NOUN

"I would get rid of the bundle if the rope breaks"

چِهن و/*čhin-o*/if break(s) , in above example we have a conditional marker و/o which is dependent upon the head of the phrase as `mark` and the verb چِهن/*čhin* is dependent on the `root` as `advcl:cond`

In Torwali, sentence can also have both morphological and syntactic marking of conditionals, by adding an optioal کو/*ko* to the right of the clause with an already present mandatory conditional marker و/o at the leftmost end of the clause.

(14)

لهات	باٹ آ	آ	و	چھن	کھے	کو
lhat	bæṭa	a	o	čhin	khe	ko
emptied	bundle.Obl.SG	me	if.COND	break	rope	if
VERB	NOUN	PRON	SCONJ	VERB	NOUN	SCONJ

Such constructions show that in Torwali, "کو/*ko*" is somewhat optional which typically serve to reinforce the conditional meaning, clarify clause boundaries, or emphasize tense/aspect/discourse nuances. Somewhat like Urdu and Pashto in the following examples (both meaning "If he had come, I would have gone"):

Urdu:

اگر وہ آتا تو میں جاتا۔
Agar voh ātā to māi jātā

Pashto:

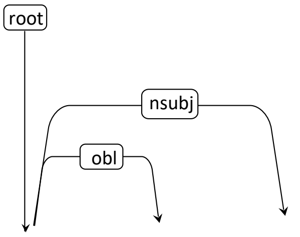
که هغه راغلی وای، نو زه تللی وم
ka haġa rāġlī wāy, no za talelī wom

4.8 Case

Based on Sir Aurel Stein's collection of three historical texts from 1926, Grierson's 1929 manuscript examines key grammatical features of Torwali. He also outlines the noun case system by identifying eight cases: nominative, accusative, ergative, instrumental, dative, ablative, genitive, and locative.

In Torwali, case suffixes are attached to nouns, which sometimes are phonologically bound and cannot stand alone. We treated the case markers in Torwali like the ergative suffix a/ ا *or e/ے* in annotation based on their phonological properties. Since the case suffixes we encountered were phonologically bound, we chose to treat them as a single token because noun+case-suffix behaved more like a single unit due to tight phonological bonding and lack of syntactic separability between the two elements. For example, as shown in (12) from section 4.7, آ-باٹ/bæṭ-a, bæṭ/bundle is noun and bæṭ-a/bundle.obl is the oblique case. Which usually is pronounced as a single unit. Other examples (15) and (16) below shows how we treated شیرے / *šire*/ in the house, and شیر مے/*šir me*/ in the house/, in the sentences below:

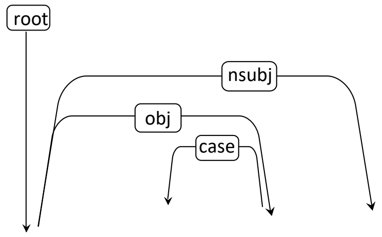
(15) *Šir-e:*



تھو	شیرے	ساجد
thu	šir-e	Sajid
is.aux	house.obl.sg	Sajid
VERB	NOUN	NOUN

"Sajid is in the house"

(16) *Šir me:*



تھو	مے	شیر	ساجد
thu	me	šir	Sajid
is.AUX	in.ADP	house	Sajid
VERB	ADP	NOUN	NOUN

"Sajid is in the house"

In (16),we treated مے /*me*/ as a postpostion based on the fact that مے/ *me*/ , is usually used in torwali as a separate word meaning "in".

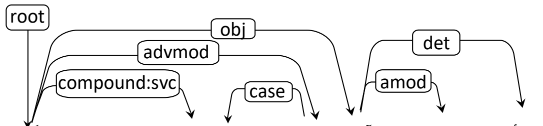
4.9 Compound verb constructions

Torwali exhibits productive compound verb formations, which involves a sizable number of verb-noun and verb-verb concatenations exemplified by the following verb-verb compound.

لار تَھل
thəl lar
throw fall

تَھل */thəl/*throw functions as the main verb and لار */lar/*fall functions as the aspectual light verb modifying the main verb. To annotate such constructions, the `compound:svc` relation is used as shown in the following example.

(17) # text = مهأ خزا آن بَهنئی وا لار تَهل



تَهل	لار	وا	بَهنئی	آن	خزا	مهأ
thəl	*lar*	*wa*	*bhə~i*	*an*	*xəza*	*mhæ*
Throw	fall	on	outside	eggs	spoiled	these
VERB	VERB	ADP	ADV	NOUN	ADJ	DET

"Throw these spoiled eggs outside"

If we jump back to example sentence from (4.3)

گأ	لار	مے	نأر	جوَؤی
gæ	*lār*	*me*	*nær*	*ǰəwəi*
go.PST.F	fall	in	dance	woman.SG

" the woman fell while dancing"

Here again, لار گأ/ *lar gae* is a compound verb where گأ /gæ act as a Feminine past auxiliary verb (from بیو /*bəyu*/ to go) and shows perfective aspect (completed action), and مے/*me*/during is a subordinating conjunction.

Similarly, for noun-verb concatenation, we used treated the verb as the head and we used the relation `compound:lvc` which is used in other Indo-Aryan languages as well, in which such construction exists. Example sentence (1) from section 4.1 shows a noun-verb complex.

4.10 Multiword tokens and reduplication

We also encountered idiomatic adverbial phrases like the one given below:

بهیس پہ بهیسا/ bhes pə bhesa/ for no reason

In this case, "بهیسا/bhesa" might have developed a pragmatic extension and used standalone for *for no reason* or *without reason.* But, "بهیس پہ بهیسا/bhes pə bhesa" as a whole is a reinforced idiom, a structure that emphasizes the meaning by repeating or echoing. The entire unit behaves as syntactically atomic and for now we have annotated and treated "بهیس پہ بهیسا/ bhes pə bhesa " and other such phrases as a multiword token with relation to the head as `advmod.`

In Torwali, reduplication is also attested, likely serving derivational or intensifying functions (e.g., pluralization, aspectual marking) and we treated them as multiword tokens. Below are some examples of such words.

/گیل میل/gel mel/bread-and-all/

/چُن چُن/čun čun/very small/

/پھٹ پھٹ/phiṭ phiṭ/pieces/

/چئ مئ/čəi məi/ tea and such/

/دستی دستی/dəsti dəsti/very quickly/

There is also verb repetition in Torwali, like the phrase "بهییل بهییل" /*bhəyel bhəyel*/ is a reduplicated verb form that functions as an adverbial phrase meaning "while sitting" or "in the midst of sitting". We treated such reduplicated verbs as a single token.

Conclusion

Torwali displays mixed characteristics, but the majority of its features are those of a fusional language, employing multiple strategies to convey grammatical and semantic information. This includes stem modification, suprasegmental changes, and reduplication, all of which are used to modify the meanings of words (Lunsford, 2001). In addition, there is extensive use of pitch and tone as grammatical markers, playing a crucial role in distinguishing between word forms and meanings.

This work presents an analysis of the basic grammatical and linguistic features of the Torwali, documented during morphosyntactic annotation and the development of a treebank. As the annotation process continues and the treebank expands, we expect to encounter additional morphosyntactic features, contributing further to our understanding of the language's structure and complexity.

Acknowledgments

This work received support from the CA21167 COST action UniDive, funded by COST (European Cooperation in Science and Technology); and from the grants "Language Understanding: from Syntax to Discourse of the Czech Science Foundation (Project No. 20-

16819X)"; and LINDAT/CLARIAH-CZ (Project No. LM2023062) of the Ministry of Education, Youth, and Sports of the Czech Republic.

We extend our thanks to Mr.Allauddin for his contributions in linguistic analysis of Torwali texts and for guiding us with his linguistic knowledge of Torwali, and to Mr. Inam Ullah for his support in sharing his valuable Torwali language data.

References

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

Ullah, Inam. 2004. ‘Lexical Database of the Torwali Dictionary.’ In The Asia lexicography conference. Chiangmai: Payap University

Lunsford, Wayne. 2001. An Overview of Linguistic Structures in Torwali, A Language of Northern Pakistan. M.A. Thesis, University of Texas at Arlington.

Grierson, George A. 1929. Torwali: An Account of a Dardic Language of the Swat Kohistan. Royal Asiatic Society. London.

Biddulph, John. 1880. *Tribes of the Hindoo Koosh*. Calcutta: Office of the Superintendent of Government Printing.

Moseley, Christopher. 2010. *UNESCO Atlas of the World's Languages in Danger*. UNESCO. Available online at: http://www.unesco.org/culture/languages-atlas.

Liljegren, H., & Akhunzada, F. (2017). Linguistic diversity, vitality and maintenance : A case study on the language situation in northern Pakistan. Multiethnica. Meddelande Från Centrum För Multietnisk Forskning, Uppsala Universitet, (36–37), 61–79. Retrieved from https://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-148722

Schuster, Sebastian and Manning, Christopher D. (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. European Language Resources Association (ELRA), Portorož, Slovenia.

Natalia Levshina. 2019. Token-based typology and word order entropy: A study based on universal dependencies. Linguistic Typology, 23(3):533–572.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Ullah, Inam (2019) "Digital Dictionary Development for Torwali, A Less-studied Language: Process and Challenges," Proceedings of the Workshop on Computational Methods for Endangered Languages: Vol. 2 , Article

Uddin, Naeem., & Uddin, Jalal. (2019). A step towards Torwali machine translation: an analysis of morphosyntactic challenges in a low-resource language. In Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages, 6-10.

Hyunji Hayley Park, Lane Schwartz, and Francis Tyers. 2021. Expanding Universal Dependencies for polysynthetic languages: A case of St. Lawrence Island Yupik. In Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas, pages 131–142, Online. Association for Computational Linguistics.

Aryaman Arora. 2022. Universal Dependencies for Punjabi. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5705–5711, Marseille, France. European Language Resources Association.

Alexey Koshevoy, Anastasia Panova, and Ilya Makarchuk. 2023. Building a Universal Dependencies Treebank for a Polysynthetic Language: the Case of Abaza. In *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*, pages 1–6, Washington, D.C.. 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. Association for Computational Linguistics.

Raverty, Henry George. *An Account of Upper and Lower Suwat, and the Kohistan, to the Source of the Suwat River; with an Account of the Tribes Inhabiting those Valleys*. Journal of the Asiatic Society of Bengal, vol. 31, no. III, 1862, pp. 227–281.

Rensch, Calvin R. 1992. Patterns of language use among the Kohistani of the Swat valley.

Sociolinguistic survey of northern Pakistan, Volume 1, Languages of Kohistan, by Calvin R. Rensch, Sandra J. Decker, and Daniel G. Hallberg. Islamabad: National Institute of Pakistan Studies and Summer Institute of Linguistics.

Legal-CGEL: Analyzing Legal Text in the CGELBank Framework

Brandon Waldon **Micaela Wells** **Devika Tiwari** **Meru Gopalan** **Nathan Schneider**
Georgetown University
{bw686, mew174, dt719, mg2293, nathan.schneider}@georgetown.edu

Abstract

We introduce Legal-CGEL, an ongoing treebanking project focused on syntactic analysis of legal English text in the CGELBank framework (Reynolds et al., 2023), with an initial focus on US statutory law. When it comes to treebanking for legal English, we argue that there are unique advantages to employing CGELBank, a formalism that extends a comprehensive—and authoritative—formal description of English syntax (the *Cambridge Grammar of the English Language*; Huddleston and Pullum, 2002). We discuss some analytical challenges in extending CGELBank to the legal domain. We conclude with a summary of immediate and longer-term project goals.

1 Introduction

There is widespread interest in the syntactic structure of legal language across multiple disciplines. For example, recent work in cognitive science has investigated how legal English differs from non-legal registers with respect to various syntactic features associated with processing difficulties (Martínez et al., 2022a,b). Modern AI research assesses the ability of artificial systems to perform legal reasoning (Guha et al., 2023, *inter alia*), which requires sophisticated understanding of complex syntactic structures found in legal documents.

There is also significant interest within legal academia and the practicing legal community: legal outcomes can hinge on a judge's reading of a single structurally ambiguous phrase in a statute or contract. Modern US legal theory (particularly the widely-adopted *textualist* framework of legal interpretation) relies on heuristics ('canons') designed to facilitate interpretation in 'hard' legal cases. For example, the Conjunctive/Disjunctive (CD) canon (Scalia and Garner, 2012, revisited in §4.1) guides interpretation of negative disjunction of the form *not A or B*. According to this canon, "'not A, B, *or* C' means 'not A, not B, *and* not C'." Linguists—including two co-authors of this paper—have at times weighed in directly through *amicus curiae* ('friend of the court') legal briefs on hard cases of textual interpretation, lending analytical insights into the syntactic as well as semantic properties of contested legal language (Champollion et al., 2023; Tobia et al., 2024, *inter alia*).

"The provisions of this section... may not be used... to **attack or defeat any title to property** after it is conveyed by the Corporation."

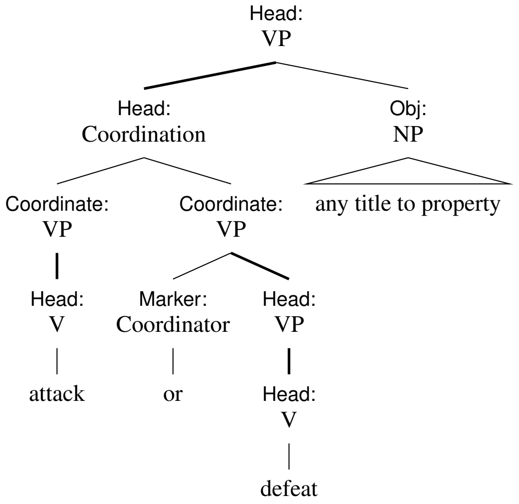


Figure 1: A portion of an annotated tree, illustrating an instance of transitive VP coordination in Legal-CGEL.

Despite this interest, there exist (to our knowledge) no sizeable gold treebanks of legal English, limiting the ability of linguists to provide grounded, quantitative insights into the grammatical properties of legal language. We aim to rectify this empirical gap with Legal-CGEL,[1] an ongoing treebanking project focused on syntactic analysis of legal English text in the CGELBank framework (Reynolds et al., 2023). Section 2 briefly recaps key properties of CGELBank, including design features which make CGELBank particularly well-suited for legal English treebanking. Section 3

[1] https://github.com/nert-nlp/legal-cgel/

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 148–153
August 28-29, 2025 ©2025 Association for Computational Linguistics

describes Legal-CGEL's development procedure and presents key statistics of the treebank in its current in-progress form. Section 4 demonstrates how Legal-CGEL enables empirical evaluation of the legal 'canons' of textual interpretation. Section 5 concludes with future goals.

2 Why extend CGELBank to legal English?

While CGELBank is a relative newcomer in the space of treebanking frameworks, it possesses advantages over more established formalisms such as Penn Treebank (PTB; Marcus et al., 1993) and Universal Dependencies (UD; de Marneffe et al., 2021) when it comes to syntactic analysis of English in general and of legal English in particular.

First, the grammar upon which it builds (the *Cambridge Grammar of the English Language*, or CGEL; Huddleston and Pullum, 2002) aims to be an exhaustive account of English syntax: across more than 1700 pages, Huddleston and Pullum (2002) ground their analysis in many hundreds of distinct synthetic data points, resulting in "the most recent comprehensive reference grammar of English, describing nearly every syntactic facet of present day Standard English" (Reynolds et al., 2023). CGELBank draws on CGEL to provide a robust description of both English constituent structure (unlike UD) and grammatical functions (unlike PTB). Its expressivity comes at the expense of cross-linguistic generalizability, which is important to other treebanking enterprises (e.g., UD) but is far less important in the context of US law. This analytical foundation is further augmented by the ongoing efforts of the CGELBank project, which evaluates and refines CGEL against naturally-occurring corpus data: CGELBank 1.0[2] analyzes 257 sentences from Twitter and the English Web Treebank.

Moreover, CGELBank is uniquely interoperable across relevant academic disciplines and the professional legal community. This is because CGEL is an authoritative formal description of English syntax familiar to lawyers and linguists alike. In the legal database WestLaw, a search of the exact string "Cambridge Grammar of the English Language" returns over 40 US state and federal cases in which the grammar is cited in a court opinion or order. This alone sets CGELBank far apart from PTB and UD, which invoke constructs that are likely unfamiliar to non-linguists in general and to the legal community in particular.

For example, unlike CGELBank, "PTB... draws heavily from particular syntactic theories like Government and Binding" (Reynolds et al. 2023: 221). By comparision, the core concepts of CGEL are couched in widely-familiar descriptive terminology and are further explicated in an undergraduate textbook (*A Student's Introduction to English Grammar*, Huddleston et al., 2022), broadening the accessibility of CGEL (and therefore CGELBank) to a more general lay audience.

3 Legal-CGEL: current status

The first iteration of Legal-CGEL focuses on US statutes, though the project may in principle be extended to other legal domains (e.g., contracts) and national contexts. To date, Legal-CGEL consists of 49 carefully-adjudicated trees of sentences drawn from the United States Code, the official codification of US federal statutes as compiled by the Office of the Law Revision Counsel (OLRC) of the United States House of Representatives. We sourced sentences of Legal-CGEL from the OLRC release point of the US Code known as Public Law 118-78,[3] which reflects the state of the US Code as of July 30, 2024. This release point is divided into 54 titles (e.g., Title 17: *Agriculture*) organized into chapters (e.g., Title 17, Ch. 24: *Honeybees*) which are further subdivided into sections (e.g., Title 17, Ch. 24, §281: *Honeybee importation*).

The OLRC maintains an XML-format digital version of the US Code, structured using the United States Legislative Markup (USLM) standard maintained by the Government Publishing Office.[4] Within the treebank, every sentence is assigned a unique identifier based on the USLM metadata of its enclosing element. To simplify navigation and cross-referencing, we added a brief, distinctive prefix to each sentence ID, e.g., `usc-039` for the 39th sentence. We restrict our analysis to the primary statutory text of the US Code; we ignore, e.g., statutory and editorial notes (which are associated with specialized USLM elements).

The 49 sentences annotated to date were hand selected to highlight a diverse set of grammatical phenomena across a range of US Code titles. The treebank currently consists of a total of 1675 lexical nodes (non-punctuation tokens) and an average of

[2] https://github.com/nert-nlp/cgel

[3] https://uscode.house.gov/download/releasepoints/us/pl/118/78/usc-rp@118-78.htm

[4] https://github.com/usgpo/uslm

POS	Phrasal Cat.	Gram. Function
479 N	618 Nom	2410 Head
277 D	445 NP	340 Mod
276 P	341 VP	314 Obj
153 V	290 PP	281 Comp
120 Adj	279 DP	276 Det
96 V_{aux}	235 Clause	121 Coordinate
55 Coordinator	126 AdjP	95 Marker
40 Sdr	55 Coordination	83 Subj
37 Adv	41 $Clause_{rel}$	26 Supplement
24 N_{pro}	39 AdvP	24 PredComp
34 *GAP*		14 Prenucleus

Figure 2: Counts for Legal-CGEL POS tags, phrasal categories, and grammatical functions. Low-frequency category and function tags are omitted from the table.

```
# sent_id = ...
# text = the Attorney General
# sent = the Attorney General
(NP
    :Det (DP
        :Head (D :t "the"))
    :Head (Nom
        :Head (N :t "Attorney")
        :Mod (AdjP
            :Head (Adj :t "General"))))
```

Figure 3: Example of the project-native `.cgel` data format, demonstrating CGELBank analysis of the noun phrase *the Attorney General*.

"Upon failure to **store or deliver to the Secretary the farm marketing excess** within such time as may be determined under regulations prescribed by the Secretary, the penalty computed as aforesaid shall be paid by the producer."

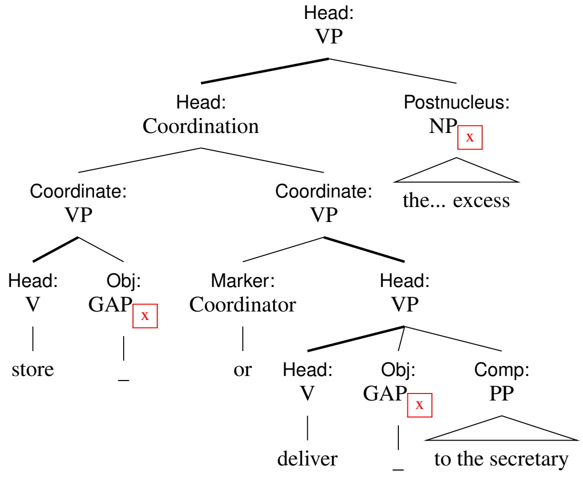


Figure 4: A nonstandard case of VP coordination.

34.2 lexical nodes per tree. A breakdown of our data by CGELBank labels (POS, phrasal category, grammatical function) is presented in Table 2.

Annotators employ ActiveDOP (van Cranenburgh, 2018), a web-based graphical treebank annotation tool which utilizes disco-dop (van Cranenburgh et al., 2016), an active learning parser. We further developed a CGELBank-specific version of ActiveDOP first reported by Reynolds et al. (2023) so that annotators could edit CGELBank trees in the project-native `.cgel` data format (Figure 3; see Reynolds et al. 2023, Sec. 5 for further discussion of the `.cgel` format).[5] Annotators manually correct automated sentence tokenizations according to CGELBank conventions (Reynolds et al., 2024); annotators also note structural ambiguities that are unresolvable out of context.

Annotations are contributed by a team of five annotators (all co-authors of the paper), including one co-developer of the CGELBank framework (NS). The remaining annotators are students and scholars of linguistics trained in CGELBank analysis. Initially, we reviewed annotations through live, team-wide discussions; however, more recent contributions were made using a GitHub-based annotation procedure (Waldon and Schneider, 2025): annotators contribute trees directly to the project GitHub repository as pull requests that are reviewed by the first and/or last author prior to acceptance. As part of this procedure, annotations are automatically visualized and validated using GitHub action scripts. The automated CGELBank validator we employ has been shown to improve inter-annotator consistency (Reynolds et al. 2023). Because adjudication proceeds over GitHub, the project repository also contains numerous discussions between project contributors. These discussions, recorded as pull request comments, can help future researchers understand the rationale behind the analytical decisions reflected in the final annotations.

Sentences of the US Code have posed analytical challenges not encountered in previous CGELBank annotation initiatives. For example, the project maintains a running list of legal terms of art (e.g., *adversary proceeding*, *due process rights*, *Attorney General*) which are to be treated as single constituents. However, some analytical decisions are suggestive of revisions to the general CGELBank annotation guidelines (Reynolds et al., 2024).

For example, CGELBank "as a rule, avoids invisibilia—but unbounded dependencies and other noncanonical word order constructions are the exception" (Reynolds et al., 2024: 32). Accordingly, for transitive VP coordinations, CGELBank treats the object as an NP complement of a coordination phrase (as in Figure 1) rather than marking the internal argument structure of the coordinated VPs with gaps coindexed to the NP.

A challenge is posed by phrases such as the one

[5] https://github.com/nschneid/activedop

found in Figure 4. For oblique dative constructions, CGELBank canonically marks rightward displacement of the direct object with a gap (e.g., *deliver* $\_{x}$ *to the secretary* [*the farm marketing excess*]$_x$). When such constructions are coordinated with simple transitive VPs (e.g., *store... the farm marketing excess*) to yield 'asymmetric' coordination structures, we made the decision to include gaps in both VPs in order to maintain consistent co-indexing across both coordinated structures and to properly represent the fact that the single displaced NP functions as the direct object of both verbs despite their different complement structures.

4 Testing canons of interpretation

In this section, we show how Legal-CGEL can provide an empirical basis on which to evaluate the textualist 'canons' of legal interpretation. In two case studies, we show that some canons encode linguistic generalizations which are readily evaluated with the help of the CGELBank framework.

Our ultimate aim is to build automated parsers of US law, to obtain quantitative estimates of how well the canons describe actual conventions of legal drafting. For now, we focus on individual trees from our gold treebank to illustrate the potential of CGELBank in two distinct use cases. In Section 4.1, CGELBank enables us to robustly characterize a class of sentences in which we expect to observe a legally-relevant semantic scope ambiguity. In Section 4.2, CGELBank provides a formal characterization of a second relevant structural ambiguity, one for which the formalism additionally expresses the range of possible disambiguations.

4.1 Conjunctive/Disjunctive (CD)

Recall from Section 1 the Conjunctive/Disjunctive (CD) canon of interpretation, which states a strong generalization of linguistic meaning: "'not A, B, *or* C' means 'not A, not B, *and* not C'."

As discussed by a group of linguists writing as *amici curiae* in *Campos-Chaves v. Garland* (Champollion et al., 2023), which concerned the interpretation of a US federal immigration statute, *not A or B* is in fact ambiguous between a 'surface scope' reading (whereby *not* takes scope over the disjunction: $\neg[A \vee B]$) and an 'inverse scope' reading (whereby *not* scopes under it: $\neg A \vee \neg B$). Champollion et al. (2023) observe that the CD canon acknowledges only the surface-scope reading; its proponents erroneously presume that logical considerations rule out alternative readings. (Scalia and Garner 2012 claim "[t]he principle that 'not A, B, or C' means 'not A, not B, and not C' is part of what is called *DeMorgan's theorem*").

"If the United States district court... determines that the surveillance was **not lawfully authorized or conducted**, it shall... suppress the evidence..."

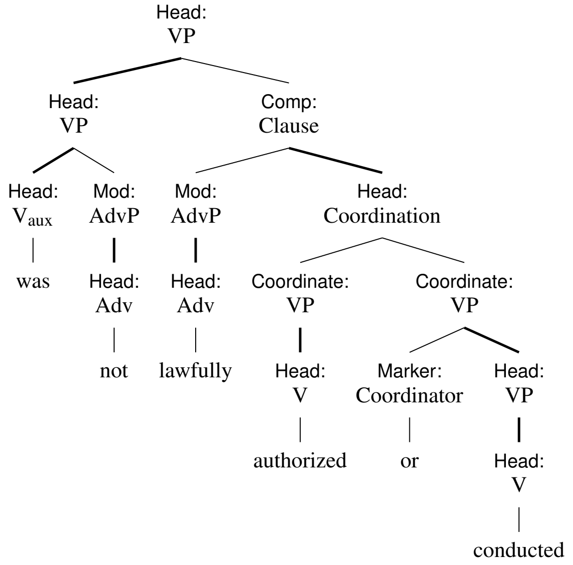


Figure 5: A narrow-scope negation identified by Champollion et al. 2023—the most plausible reading is that evidence is suppressed if surveillance is unlawfully authorized or unlawfully conducted.

The CD canon states an empirically-verifiable hypothesis regarding linguistic interpretation, one that Champollion et al. (2023) problematize by presenting examples of the inverse-scope reading within the US Code. Legal-CGEL includes a CGELBank analysis of one such sentence, as illustrated in Figure 5. The tree explicitly models negative disjunction as a particular structural interaction of negation (*not*) and the coordination (*authorized or conducted*). Of course, the tree does not specify how the relevant scope ambiguity is actually resolved (a matter we leave to careful human annotation). However, for a large dataset of CGELBank-parsed trees, a structure-based query would allow us to efficiently isolate the space of sentences in which we expect the ambiguity to manifest (cf. linear searching methods such as regex, which would likely yield many false positives: e.g., [***not*** *lawfully authorized*] ***or*** [*haphazardly conducted*]).

4.2 Nearest Reasonable Referent (NRR)

Like the CD canon, the Nearest Reasonable Referent (NRR) canon is formulated as a linguistic generalization. The NRR canon states that "[w]hen the syntax involves something other than a parallel

“Any alien whose permanent resident status is terminated under paragraph (1) may **request a review of such determination in a proceeding** to remove the alien.”

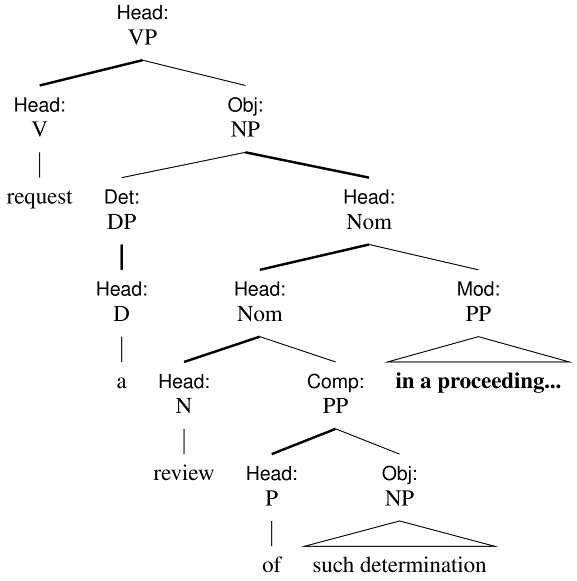


Figure 6: An example of ambiguous PP attachment in the treebank.

series of nouns or verbs, a prepositive or postpositive modifier normally applies only to the nearest reasonable referent” (Scalia and Garner, 2012).

Here, too, legal treebanking can facilitate empirical evaluation of a legal interpretative principle. As a formalism that captures both constituency structure and functional relationships between constituents, CGELBank provides an ideal basis for modeling the structural dependencies that underlie the NRR canon’s predictions regarding prepositive and postpositive modifier scope.

This aspect of the framework is illustrated in Figure 6, which partly reproduces a structurally ambiguous sentence found in the treebank. On the NRR-consistent reading (the one presented in Figure 6), a noncitizen alien requests review of the termination of their permanent resident status, and that review occurs in a removal proceeding. On a second reading, a noncitizen alien makes the request in a removal proceeding. This second reading would be reflected with a higher attachment site of the PP modifier *in a proceeding...*, i.e., at the VP level. In this case, the annotator marked the presence of this structural ambiguity and provided a brief characterization of it as part of the annotation.

5 Conclusion and future directions

We have introduced and motivated Legal-CGEL, an ongoing legal treebanking initiative in the CGELBank framework. In addition to expanding the treebank to many more sentences, we plan to measure inter-annotator agreement to assess the consistency of our annotation conventions. Longer-term, we plan to build and evaluate automated CGELBank parsers, which will enable large-scale analysis of the syntactic properties of our target domain.

6 Acknowledgments

We thank Kevin Tobia, Sara Frug, and Sylvia Kwakye for their helpful guidance regarding the structure and organization of the US Code. We also thank Brett Reynolds for his insightful comments on many of the treebank entries. Lastly, we gratefully acknowledge the feedback of three anonymous TLT reviewers. This research was supported in part by NSF award IIS-2144881 (to NS).

References

Lucas Champollion, Brandon Waldon, Masoud Jasbi, Willow Parks, and Cleo Condoravdi. 2023. Brief for amici curiae Lucas Champollion, Brandon Waldon, Masoud Jasbi, Willow Parks, and Cleo Condoravdi in support of noncitizens Campos-Chaves, Singh, and Mendez-Colín. *Campos-Chaves v. Garland*, Docket No. 22-674.

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

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Aditya K, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 44123–44279. Curran Associates, Inc.

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

Rodney Huddleston, Geoffrey K. Pullum, and Brett Reynolds. 2022. *A Student’s Introduction to English Grammar*, 2nd edition. Cambridge University Press, Cambridge, UK.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Eric Martínez, Francis Mollica, and Edward Gibson. 2022a. Poor writing, not specialized concepts, drives processing difficulty in legal language. *Cognition*, 224:105070.

Eric Martínez, Francis Mollica, and Edward Gibson. 2022b. So much for plain language: An analysis of the accessibility of United States federal laws (1951-2009). In *Proc. of the Annual Meeting of the Cognitive Science Society*, volume 44, pages 297–303.

Brett Reynolds, Aryaman Arora, and Nathan Schneider. 2023. Unified syntactic annotation of English in the CGEL framework. In *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 220–234, Toronto, Canada. Association for Computational Linguistics.

Brett Reynolds, Nathan Schneider, and Aryaman Arora. 2024. CGELBank Annotation Manual v1.1. *Preprint*, arXiv:2305.17347.

Antonin Scalia and Brian A. Garner. 2012. *Reading Law: The Interpretation of Legal Texts*. Thomson West, Eagan, Minnesota.

Kevin Tobia, Nathan Schneider, Brandon Waldon, James Pustejovsky, and Cleo Condoravdi. 2024. Brief for professors and scholars of linguistics and law as amici curiae in support of petitioners. *Bondi v. VanDerStok*, Docket No. 23-852.

Andreas van Cranenburgh. 2018. Active DOP: A constituency treebank annotation tool with online learning. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 38–42, Santa Fe, New Mexico. Association for Computational Linguistics.

Andreas van Cranenburgh, Remko Scha, and Rens Bod. 2016. Data-oriented parsing with discontinuous constituents and function tags. *Journal of Language Modelling*, 4(1):57–111.

Brandon Waldon and Nathan Schneider. 2025. A GitHub-based workflow for annotated resource development. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX)*, Vienna, Austria. Association for Computational Linguistics. To appear.

Status of morphosyntactic features
Illustration with written and spoken French UD treebanks

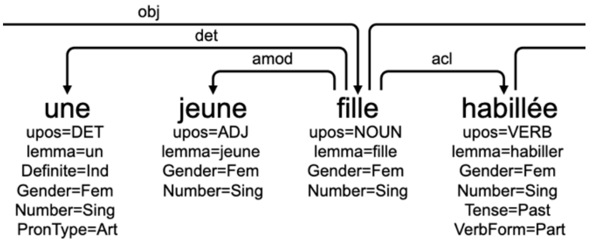
Sylvain Kahane*Δ   Bruno Guillaume◊   Léna Brun*   Simeng Song*
*Modyco, Paris Nanterre Université & CNRS   ΔInstitut Universitaire de France
◊Université de Lorraine, CNRS, Inria, LORIA, France

Abstract

Morphosyntactic features used in UD treebanks have different status. If most of them correspond to values of inflectional morphemes, some describe lexical subclasses or are just conventional names of (polysemic) morphemes. Syncretism is also a challenge, because exact values are only deductible from contextual information. We propose an attempt at clarification and an implementation in the treebanks of written and spoken French.

1 Introduction

In Universal Dependencies (UD) annotation scheme for syntactic treebanks, syntax is encoded by relations between words, while morphosyntax is encoded by features on words (de Marneffe et al. 2021). For instance, in Fig.1, the noun *fille* 'lady' has three dependents: the determiner (det) *une* 'an', the adjectival modifier (amod) *jeune* 'young', and the past participle *habillée* (*en noir*) 'dressed (in black)', analyzed as an adjectival clause (acl). Each of the four words bears features indicating their POS (upos), their lemma, as well as morphosyntactic features, such as Gender, Number, Tense, etc.



a   young   lady   dressed (in black)

Fig. 1. Extract from UD_French-Rhpsodie@2.16

Morphosyntactic features can have different status. For instance, in French, adjectives agree with nouns in gender (and number): gender on French adjectives is an agreement morpheme, marking the relation with a noun, while gender on nouns is a pure lexical feature, an inherent feature of the lexeme, triggering the agreement of adjectives and determiners (Mel′čuk 1993:261, 2006; Corbett 2022; McCarthy et al. 2018). Moreover, if French adjectives always agree in gender, many of them have a common form for feminine and masculine and it is unclear whether they should bear a Gender feature. This phenomenon, which is already important in written French, become widespread in spoken French, where, for instance plural nouns are written with a *s* at the end, which is phonologically realized only in the rare case of the optional liaison with a following adjective beginning with a vowel (*des femmes* /de fam/ 'women'; *des femme***s** *illustres* /de fam **z** ilystr/ 'famous women').

Syncretism (Corbett 2011) is also a source of numerous discrepancies. When a form corresponds to several values of features, do we annotate the set of values associated to the form or the value that can be inferred from the context? For instance, if the English verbal form *thinks* clearly deserves the features Number=Sing, Person=3, what should be done with *think*? Currently, UD treebanks for English give the values inferred by the context, but these values do not have the same status as the value for *thinks*. As noted by Malaviya et al. (2018), the adjective *refrescante*, which is not inflected for gender in Spanish and Portuguese, has a Gender feature in Portuguese UD treebanks, but not in Spanish ones.

A last problem is posed by the traditional names of inflectional morphemes. UD morphosyntactic features are supposed to be universal features (de Marneffe et al. 2021), that is, comparative concepts, as opposed to language-specific categories (Haspelmath 2011). Romance and Germanic languages have two participles that are traditionally called the present and the past participles. Accordingly, particles in UD treebanks

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 154–159
August 28-29, 2025 ©2025 Association for Computational Linguistics

for English or French have a feature Tense=Pres or Tense=Past on every participle (see *habillée* 'dressed' in Fig. 1), while aspectual features, such as Aspect=Imp or Aspect=Perf, would have been much more motivated from the universal point of view. Only the feature Voice=Pass appears on past participles used in passive constructions.

In this paper we propose to distinguish four types of morphosyntactic features: plain features, such as Number, are used for the values of inflectional morphemes of the word, Number[ctxt] is used for the value inferred from the context, Gender[lex] is used for lexical features, and Tense[denom] is used for denominative features. [1]

In the following sections, we start by some examples (Section 2), then we discuss the formalism we use (Section 3). Section 4 presents possible problems of delimitation between lexical and inflectional features and Section 5, between presence and absence of a feature. Sections 6 and 7 are dedicated to the implementation of our annotation in treebanks of written and spoken French. Our conclusion (Section 8) comes back to the relevance of such an annotation for linguistic studies.

2 First examples

In the French example (1), the noun *exercice* 'exercise' is masculine; the indefinite article *un* agrees with it, while the adjective *utile* 'useful', which does not vary in gender, inherits the masculine from the context.

(1)	*un*	*exercice*	*utile*
	an	exercise	useful
	Gender=Masc		Gender[ctxt]=Masc
		Gender[lex]=Masc	

In Russian, nouns vary in case and can have six different singular forms. The noun *žurnal* 'magazine' has the same form in the nominative and accusative cases. We propose to distinguish the case value associated to the form (type-level) from the case value given by the context (token-level):

(2)	*novyj*	*žurnal*
	new	magazine
	Case=Nom	Case=Nom,Acc
		Case[ctxt]=Nom

The English verbal form *arrived* can be a preterit or past participle. In *she has arrived*, the word will receive the following features:

(3) *arrived*
VerbForm[ctxt]=Part, Tense[denom]=Past
Aspect[ctxt]=Perf

They indicate that, in this context, the form is a participle with a perfective aspectual value, which is denominated the *past* participle. Because the past participle has the same form as the preterit, we can indicate that VerbForm is a contextual feature using a feature VerbForm[ctxt], or we can consider that it is a case of homonymy and use the feature VerbForm without extension.

3 Formalization

We use the notation of layered features, which has been introduced for another purpose, when a word has two features of the same type (https://universaldependencies.org/u/overview/feat-layers.html). For instance, auxiliaries in Basque can agree with several arguments: the form *dute* marks the agreement in number and person with an absolutive and an ergative argument:

(4) *dute*
upos=AUX, lemma=edun,
VerbForm=Fin, Mood=Ind,
Number[abs]=Sing, Person[abs]=3
Number[erg]=Plur, Person[erg]=3

It is not the only possible formalization, but this one is already integrated in the query language of main query systems, such as Grew-Match (Guillaume 2021).[2] Note that we cannot exclude that a layered feature is syncretic and to have a feature such as Number[abs][ctxt].

We can also remark that the layer [psor] that has been introduced for personal determiners that agree both with their governor and with the possessor is in fact a lexical feature. The current annotation for the German possessive *seine* 'his.FEM' in (5a) can be replaced by (5b):

(5) a. *seine*: Number=Sing, Gender=Fem,
Number[psor]=Sing, Gender[psor]=Masc
b. *seine*: Number=Sing, Gender=Fem,
Number[lex]=Sing, Gender[lex]=Masc

[1] The Leipzig Glossing Rules also advocate a particular convention for lexical features: "Inherent, non-overt categories such as gender may be indicated in the gloss, but a special boundary symbol, the round parenthesis, is used."

[2] Because brackets are special symbols in Grew, the request uses a double underscore: pattern { X [Number__erg] }. See e.g. https://universal.grew.fr/?custom=684da8a0a4075.

4 Inflectional vs lexical features

Lemmas and the repartition between lexical and inflectional features both depend on what we consider as inflectional paradigms. In French, nouns denoting persons or animals can have a masculine and a feminine form: *un instituteur* 'a teacher. MASC', *une institutrice* 'a teacher.FEM'. Two choices are possible:

(6) a. lemma=institutrice, Gender[lex]=Fem
b. lemma=instituteur, Gender=Fem

Following Mel'čuk (2000), the first solution has been chosen for French treebanks.

Another case where it can be difficult to decide what the inflectional paradigms are is illustrated by pronouns. Unlike nouns, French personal pronouns have different forms in subject, object and oblique positions. The traditional analysis is to consider that they vary in case. We consider that we have different pronouns for 1st and 2nd person singular and plural. The lemma is the emphatic form, which is the only form that can stand alone (the form is also used after a preposition). See (7).

(7) a. *je* '1SG.NOM',
lemma=moi, Number[lex]=Sing,
Person[lex]=1, Case=Nom
b. *me* '1SG.ACC|DAT',
lemma=moi, Number[lex]=Sing,
Person[lex]=1, Case=Acc,Dat

We have considered that personal determiner *mon* 'my' is not part of the paradigm and has its own lemma, because it varies in gender and number. The personal pronouns for 3rd person pose an additional problem. The feminine and masculine forms are different for emphatic (*elle, lui, elles, eux*), nominative (*elle* 'she', *il* 'lui', *elles, ils* 'they'), singular accusative (*la* 'her' vs *le* 'him'), but gender is neutralized in plural accusative (*les* 'them') and dative (*lui* 'to her/him', *leur* 'to them'). Moreover, the singular and plural forms are morphologically related and we decided to have the same lemma (even if it was not the choice before v2.16).

(8) *elle*, lemma=lui,
Person[lex]=3, Number=Sing,
Gender=Fem, Case[ctxt]=Nom

The genitive form *en* is currently analyzed as a separate lemma, because personal pronouns of 1st and 2nd person do not have a genitive and *en* is not related morphologically to other 3rd person pronouns.

A contrast between lexical and inflectional features is illustrated by simple vs complex verbal forms. Compare Haitian Creole and French.

(9) a. *lavi m pra chanje*
life my will change
'my life will change'
b. *ma vie changera*
my life will_change

French has a morphological future, while Haitian Creole has a separate auxiliary for future, like English. In the Haitian Creole treebank (Kahane et al. 2024), a feature Tense=Fut has been attached to the marker, but it is clearly a lexical feature:

(10) a. *pra*: upos=AUX, Tense[lex]=Fut
b. *changera*: upos=VERB, Tense=Fut

In English UD treebanks, modals have a VerbForm=Fin feature, but this feature is a lexical feature because modals do not inflect but they can only be used in finite clause:

(11) *must*: upos=AUX, VerbForm[lex]=Fin

In the same way, definiteness on article is a lexical feature: *the*, Definite[lex]=Def. (The article also has a feature PronType=Art, but such a feature is lexical by nature, and we don't need to add [lex] in such a case.)

French has a past tense, called *passé composé*, which is built like English present perfect, but is semantically more similar to the preterit. In this case also, the auxiliary can be considered as a lexical marker of the past, the inflection of the lexical verb being imposed by the auxiliary and being part of the semantics of the auxiliary.

(12) *elle est* *venue*
VerbForm=Fin VerbForm=Part
Tense=Pres Tense[denom]=Past
Tense[lex]=Past
'she came'

Such an analysis is not very different from the analysis we can do for Haitian Creole, where the auxiliary *te* is Tense[lex]=Past and the lexical verb is invariable. But in the case of French, this analysis allows us to indicate that the complex verbal form is past, even if the tense of the auxiliary is present.

5 When to annotate a feature

One question is when to annotate a feature. For instance, the French definite determiner has three form: *le* ‘the.SG.MASC’, *la* ‘the.SG.FEM’ and *les* ‘the.PL’. For the syncretic plural, do we want to have features Gender or Gender[ctxt]? A feature Gender=Fem,Masc could seem useless, but it can indicate that *les* is a form of a lemma that can vary in gender and contrast it with an adjective such as *utile* ‘useful’ that is not inflected in gender. And when *les* is combined with a noun, do we want to add a feature Gender[ctxt]?

Because a majority of French adjectives vary in gender, we have decided to add a feature Gender[ctxt] for adjectives that are not inflected in gender. But it is clear that such features are not very useful and could be omitted. Nevertheless, UD treebanks are full of such features. For a case of syncretism such as Russian *žurnal* ‘magazine’ it is more interesting to indicate that the form is Case=Nom,Acc, because this form contrasts with other forms for dative or locative. And because UD annotation is token-based, it also makes sense to indicate in particular contexts whether it is Case[ctxt]=Nom or Case[ctxt]=Acc.

For English verbal form such as *think*, it is complicated, because the form can correspond to infinitive or present tense and in present tense it can correspond to any number or any person, with the exception of the combination Number=Sing, Person=3. This cannot easily be indicated in the features. Moreover all English infinitive forms will always have VerbForm[ctxt]=Inf, because it is not possible to know that they are infinitive without the context. This is a general property of English morphology and it seems not necessary to indicate it for each occurrence of a verb. In English, many forms are polycategorial, such as *love*, which can be a verb or a noun. The conllu encoding does not allow us to have upos[ctxt], but it is not sure that we want to indicate such syncretisms.

In French, past participles of transitive verbs vary in number and gender and agree with their subject when they are passive forms and with their object when it is placed before the verb. But past participles of intransitive verbs are invariable. It is not always easy to decide whether a verb is transitive or intransitive and for the sake of simplicity, all past participles have features Number and Gender.

6 Annotation of written French

One of our main motivations to distinguish contextual vs overt values of features was the fact that many adjectives in French do not inflect in gender. It was easy to make this distinction because there are resources indicating whether each adjective inflects in gender or number, such as the Lefff (Lexique des formes fléchies du français ‘Lexicon of inflected forms of french’) (Sagot 2010). A Grew script (Guillaume 2021) based on Lefff has been applied on French-GSD (Guillaume et al. 2019). On the 23817 adjectives of the corpus, 16949 occurrences (71%) (for 2472 lemmas) were covertly marked for gender and number, but 6796 (28%) (for 1124 lemmas) were only marked for number and receive a feature Gender[ctxt], 975 (4%) (for 157 lemmas) were not marked for number at the masculine and receive a feature Number[ctxt], and 72 occurrences (0.3%) (for 22 lemmas) are from invariable adjectives:

- Most common adjectives without gender inflection: *autre* ‘other’, *même* ‘same’, *jeune* ‘young’, *propre* ‘proper, clean’, *politique* ‘political’ …
- Most common adjectives with unique form at the masculine: *français* ‘French’, *nombreux* ‘numerous’, *anglais* ‘English’, *vieux* ‘old’ …
- Most common invariable adjectives: *super* ‘super’, *standard* ‘normal’, *arrière* ‘back’, *cool* ‘cool’ …

A multilingual lexicon such as UniMorph (Sylak-Glassman et al. 2015) could allow us to do the same thing for other languages.

Some determiners are lexically singular (*chaque* ‘each’, Number[lex]=Sing) or plural (*pusieurs* ‘several’, Number[lex]=Plur). Articles vary in number and gender but have a syncretic form for plural. Beyond a vowel, the definite article and possessive determiners have a different form. As shown in (13), the masculine form of the possessive is used before a vowel whatever the gender of the noun.

(13)	*mon*	*étoile*
	Gloss=my	Gloss=star
	Gender=Masc	Gender[lex]=Fem
	Gender[ctxt]=Fem	

Numerals are interesting. They are lexically plural when they are used as determiners/cardinals, but they are singular when they are used as proper nouns:

(14) *2025 est une année très chaude*
2025 is a year very hot
upos=NUM
ExtPos=PROPN
Number[lex]=Sing

The French treebanks have some denominative features, such as Tense[denom] for participles (see Section 2). The Tense=Imp feature for *imparfait* tense is another example of denominative feature. We propose to replace it by Tense[denom]=Imp, Tense=Past, Aspect=Imp (imperfective past).

7 Annotation of spoken French

If the corpus is a spoken corpus, we must annotate the morphosyntactic properties of the spoken form and not of its written transcription. We think that it is important to state this, because it is not what was done in spoken French UD corpora before we started this study.

The question is delicate in French, because orthography marks a lot of things that are not pronounced. For instance, plural on the majority of nouns is marked by a grapheme *s*, which would only be audible if a liaison with a following adjective beginning with a vowel is realized and the liaison is almost never attested in spontaneous speech. In consequence, we consider that nouns in spoken French have no number, except for a small set of nouns finishing in *-al* or *-ail*, which have a plural in *-aux* /o/: *un cheval* /œ̃ ʃəval/ 'a horse', *des chevaux* /de ʃəvo/ 'horses'. Among the 5195 nouns in French-Rhapsodie (Lacheret et al. 2019), we have only 49 occurrences (for 9 lemmas) of such nouns.

Adjectives have also a plural marked by a grapheme *s*, but the number is not marked on adjectives in most cases and is only contextual (15a. For prenominal adjectives the liaison with a following noun starting with a vowel is obligatory and the plural will be marked in this case (15b).

(15) a. *des oiseaux très petits*
de zwazo trɛ pəti
Number=Plur Number[ctxt]=Plur
'very small birds'
b. *des petits oiseaux*
de pəti **z**wazo
Number=Plur Number=Plur
'small birds'

If the gender remains marked for a majority of adjectives (*vert* /vɛʁ/ 'green.MASC', *verte* /vɛʁt/ 'green.FEM'), it is no longer marked for adjectives finishing by a vowel (*joli* /ʒoli/ 'nice', written *jolie* /ʒoli/ in the feminine form), which also concerns past participles. In some dialect such as Belgian French, the final vowel of feminine forms such as *jolie* is lengthen, but it is not the case in the spoken corpora currently in UD.

We can also note that in spoken French, the singular present and imparfait forms of almost all verbs are similar. In consequence, for these verbs, Person is only contextual. Moreover, for most verbs the 3rd person plural is also similar, which means that Number is also contextual. In other words, only the 1st and 2nd person plural are marked. Moreover, the 1st person plural is rarely used: only 2 occurrences of *nous* 'we' subject for 755 occurrences of the indefinite pronoun *on* 'one' in French-ParisStories (Kahane et al..

In conclusion, without taking into account the specificity of spoken data and differentiating the contextual values, the treebank would have been completely misleading concerning number and gender marking.

8 Conclusion

The distinction between inflectional, lexical, and denominative features allows us to clarify the status of morphosyntactic features in UD treebanks. If we study Tense in English (without any knowledge of the language), we would have strange results due to the Tense feature on participles and the absence of lexical feature on auxiliaries would give us that idea that the language has no future.

It is also very useful for linguists exploiting the treebanks to know whether a feature is overt or it has been inferred from the context. Without such an annotation it is not possible to evaluate the range of a given feature. For instance, in French, the subject position is marked by the preverbal position, the agreement of the verb in person and number, and case on personal pronouns, but without a precise annotation it would not be possible to know which features are really effective. Same thing for the range of the noun-adjective agreement in French.

Our proposition of a more precise annotation of morphosyntactic features is a first attempt in UD treebanks and it will certainly evolve in the future. But we hope that such annotation will spread in treebanks of other languages, allowing a more accurate comparison between languages.

References

Corbett, Greville G. 2011. The penumbra of morphosyntactic feature systems. *Morphology* 21.2: 445-480.

Corbett, Greville G. 2012. Canonical morphosyntactic features. In Dunstan Brown, Marina Chumakina, and Greville G. Corbett (eds), *Canonical Morphology and Syntax*, 48-65.

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

Guillaume, Bruno. 2021. Graph Matching and Graph Rewriting: GREW tools for corpus exploration, maintenance and conversion. In *Proceedings of EACL: System Demonstrations*.

Guillaume, Bruno, Marie-Catherine de Marneffe, and Guy Perrier. 2019. Conversion et améliorations de corpus du français annotés en Universal Dependencies. *Traitement Automatique des Langues* 60 (2), pp.71-95.

Malaviya, Chaitanya, Matthew R. Gormley, Graham Neubig. 2018. Neural factor graph models for cross-lingual morphological tagging. In *Proceedings of the 56th ACL*, 2652-2662.

Haspelmath, Martin. 2010. Comparative concepts and descriptive categories in crosslinguistic studies. *Language* 86.3: 663-687.

Kahane, Sylvain, Claudel Pierre-Louis, Sandra Jagodzińska, Agata Savary (2024). The first Haitian Creole treebank. In *2nd UniDive Workshop*, Naples.

Kahane S., Caron B., Gerdes K., Strickland E. (2021) Annotation guidelines of UD and SUD treebanks for spoken corpora: A proposal. In *Proceedings of 19th international conference on Treebanks and Linguistic Theories (TLT)*, SyntaxFest, ACL.

Lacheret-Dujour, Anne, Sylvain Kahane, Paola Pietrandrea (eds) (2019), *Rhapsodie – A Prosodic and Syntactic Treebank for Spoken French*, John Benjamins, Amsterdam.

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

McCarthy, Arya D., Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2018. Marrying Universal Dependencies and Universal Morphology. In *Proceedings of the Second Workshop on Universal Dependencies (UDW)*, 91-101.

Mel′čuk, Igor A. 1993. *Cours de morphologie générale: Introduction et première partie: le mot.* Presses de l'Université de Montréal.

Mel'čuk, Igor A. 2000. Un FOU/une FOLLE: un lexème ou deux?. *Lexique, syntaxe et sémantique. Mélanges offertes à Gaston Gross à l'occasion de son soixantième anniversaire* [BULAG, numéro hors série], 95-106.

Mel′čuk, Igor A. 2006. *Aspects of the Theory of Morphology*. Vol. 146. Walter de Gruyter.

Sagot, Benoît. 2010. The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French. In *7th international conference on Language Resources and Evaluation (LREC)*.

Sylak-Glassman, John, Christo Kirov, David Yarowsky, and Roger Que. 2015. A Language-Independent Feature Schema for Inflectional Morphology. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL)*, 674-680.

Author Index