UDW 2021

Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)

Proceedings

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

Order copies of this and other ACL proceedings from:

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

ISBN 978-1-955917-17-9

Preface

Universal Dependencies (UD) is a framework for cross-linguistically consistent treebank annotation that has so far been applied to over 100 languages (http://universaldependencies.org/). The framework is aiming to capture similarities as well as idiosyncrasies among typologically different languages (e.g., morphologically rich languages, pro-drop languages, and languages featuring clitic doubling). The goal in developing UD was not only to support comparative evaluation and cross-lingual learning but also to facilitate multilingual natural language processing and enable comparative linguistic studies.

The goal of the UD workshop is to bring together researchers working on UD, to reflect on the theory and practice of UD, its use in research and development, and its future goals and challenges.

The Fifth Workshop on Universal Dependencies (UDW 2021) is, like its third edition before, part of SyntaxFest, which co-locates four related but independent events:

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

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

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

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

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

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

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

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

December 2021

Program co-chairs

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

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

Local Organizing Committee of the SyntaxFest

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

Program Committee for the Whole SyntaxFest

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

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

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

Additional Reviewers

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

Table of Contents

Formae reformandae: for a reorganisation of verb form annotation in Universal Dependencies illustrated by the specific case of Latin

Flavio Massimiliano Cecchini
Università Cattolica del Sacro Cuore / Largo Gemelli 1, 20123 Milan, Italy
`flavio.cecchini@unicatt.it`

Abstract

Nonfinite verb forms are a crosslinguistically widespread phenomenon that poses a challenge to universal annotation formalisms like Universal Dependencies (UD), often clashing with traditionally established, language-specific conventions and terminologies. This paper, using Latin as a concrete case study, aims to give a survey on the `VerbForm` feature distribution among UD treebanks and to suggest a restructuring thereof in a universal perspective.

1 Introduction

The project of Universal Dependencies (UD) aims to harbour syntactically and morphologically annotated treebanks of any language, on the basis of universal, crosslinguistically valid tagsets for parts of speech (UPOS), morpholexical features and syntactic dependency relations (de Marneffe et al., 2021). The latest release, v2.9 of November 2021 (Zeman et al., 2021), of its second version (Nivre et al., 2020) sees 122 languages with at least one treebank each for a total of 217 treebanks, among which four Latin treebanks, variously distributed on diachronic, diastratic and diaphasic axes: Classical and Late Latin works, from prose (e. g. *De bello gallico* by Caesar), poetry (e. g. *Metamorphoses* by Ovid) and religious sources (e. g. Bible) in Perseus (Bamman and Crane, 2011) and PROIEL (Eckhoff et al., 2018), Early Medieval notarial Latin from Tuscia in the Late Latin Charter Treebank (LLCT) (Korkiakangas and Passarotti, 2011; Cecchini et al., 2020a), polished Late Medieval Latin of philosophical-theological texts by Thomas Aquinas in the Index Thomisticus (IT-TB) (Passarotti, 2019; Cecchini et al., 2018) and treatises (e. g. *De vulgari eloquentia*), poetry and personal correspondence by Dante Alighieri in UDante (Tavoni, 2011; Cecchini et al., 2020b), for a total of nearly 1 million tokens. Together, they testify the livelihood of Latin as a political, scientific and literary *lingua franca* in the centuries, well up into the modern age, that followed the fall of the Western Roman Empire, where it was the administrative and everyday language of a continent-spanning dominion (Waquet, 2001; Clackson and Horrocks, 2007; Leonhardt, 2009).

Variation over the whole of UD treebanks is even greater, most importantly with respect to linguistic phenomena and grammatical traditions that have to converge into the universal formalism. This does not always take place without conflicting or incoherent annotational choices and styles, both inter- and intralinguistically: these often arise, on the one side, from the direct translation of language-specific conventional denominations into misleadingly homonymous UD labels, and on the other side from some unfortunate naming choices in UD itself borrowed from historical, Latin-influenced traditional grammars of (mostly) European languages. This paper focuses on one of such "front lines": the values of the morpholexical feature `VerbForm`, which in UD defines the different kinds and behaviours of forms in a verbal paradigm, and in particular of those values for nonfinite forms. While the discussion is centered around the UD formalism, it is also of a more general nature about the tension in the identification of language-specific and universal classes,[1] and the paper also puts forward new criteria for the annotation of Latin nonfinite verb forms as a sort of practical example for its conclusions. We notice that, especially

[1] On this tension, cf. the discussion in (Croft, 2001, §2) about the identification of universal parts of speech, and in (Haspelmath, 2009) about terminology for morphological case.

from a terminological point view, Latin has a particular position in this context, given the wide-reaching authority of its historical grammatical works.

In §2, the framework of this paper is briefly expanded upon; in §3, the particular case study of the supine in UD is detailed for Latin and other languages; in §4, a concise survey of VerbForm values' distribution in UD is given, commented in more generic terms in §5; in §6, a new system for Latin nonfinite VerbForms in UD is proposed; finally, §7 closes this work with some minor proposals.

2 Finite and nonfinite verb forms and VerbForms

In UD, inflectional, verbal features include VerbForm, glossed as "form of verb or deverbative".[2] At a universal level, the definition of VerbForm is thus left somewhat underspecified and vague, and the single values for this feature are only sketchily illustrated by few examples. A first fundamental distinction is between the value Fin and all others. The notion of "finite" verb form is not unquestionable, as it is originally anchored in the Latin verbal system: there, it is a verb form expressing an agreement in person (Person) with the subject (therefore "conjugated" with it, from *coniugo* 'join together'), a tense (Tense) and a mood (Mood), and is the only kind of VERB form that can stand as head of an independent, unmarked clause without auxiliaries (AUX). This definition is not applicable crosslinguistically (Koptjevskaja-Tamm, 1994), as can already be seen by the world-wide distribution of verbal person marking (Siewierska, 2013).[3] We will however refrain from attempting a direct universal definition of "finite" verb form, due to the non-trivial connected issues that go beyond the scope of this paper, and will leave it as a "primitive notion", contenting ourselves of its operative definition for Latin, for which it may make sense. Still, we note that a characterisation of finiteness "by negation" emerges from the identification of cardinal nonfinite forms in §5: then, finite forms are all the others. In a typological perspective, this conclusion would actually suggest to abandon the notion of "finiteness" altogether, as it does not so much correspond to an actual linguistic category as it is a language-specific co-occurrence of factors (in Latin, the expression of person, tense and mood being compulsory for main, independent predicates).[4] So, we observe a correlation between finite forms and independent clauses on one side, and nonfinite forms and embedded or subordinated clauses, see (Pinkster, 1990, §7, §8), on the other. The latter are situated in a grey area where verbs assume morphosyntactic behaviours which are usually ascribed to other parts of speech (see §5), and this characteristic makes nonfinite forms difficult to pinpoint, hence the many nonfinite values for VerbForm in UD in contrast to only one finite value Fin, and the even more detailed record of cases and denominations by traditional grammars.

Nonfinite forms are a complex phenomenon that warrants attention and that so to speak puts a crosslinguistic annotation formalism like UD to the test, hence they have been chosen as the main subject of this paper. Further, they are significantly present in Latin, so that Latin itself can serve as a valid proving ground for claims and proposals on the matter.

3 The supine as case study of a nonfinite VerbForm

The implementation of the VerbForm label Sup for the so-called supine across UD treebanks is a very specific, but nonetheless good case study for the misunderstandings that arise when confronting a crosslinguistic formalism like UD with those of traditional grammars of single languages, and for the confusion of focus between particular and more general phenomena at different annotational layers (cf. §4).

[2] In the following the general reference is the documentation found on the UD website (https://universaldependencies.org/), in particular the resumptive page for all morpholexical features (https://universaldependencies.org/ext-feat-index.html) and the page about VerbForm (https://universaldependencies.org/ext-feat-index.html#verbform), with all related universal and language-specific documentation. All data is also retrievable through each treebank's hub page, and queries on them can be performed by means of different tools, e. g. online with Grew (Guillaume, 2021).

[3] Hence we note that this terminological choice is unfortunate from a universal point of view, since it cannot be based on the original Latin notion of *(in)finitus* '(in)definite (form)'; cf. §4 about other VerbForm denominations.

[4] We note that a similar reasoning might be put forth, at least for Latin, for the "positive" degree (Degree=Pos in UD) of an adjective or similar element, since its definition actually corresponds to the *absence* of a comparative or absolutive degree.

To address the issue, we will first give a rather detailed presentation of the Latin supine, proposing a new, more typologically grounded way to annotate it, before turning to so-called supines in other languages and their representation in UD.

3.1 The Latin supine

In traditional Latin grammars, the supine form, or "mood",[5] is described among others as "a verbal abstract of the 4th declension [. . .], having no distinction of tense or person" (Greenough et al., 2014, §508)[6] or more explicitly as "a defective verbal noun" (Barbieri, 1995, §150).[7] In such works, cf. e. g. (Palmer, 1988, p. 324f.), it is noted that the supine only appears in the singular number, either in the accusative or ablative case (but dative might also be attested), assigning to the former an alleged "active" and to the latter an alleged "passive" voice, however on no clear morphosyntactic grounds. Its use is limited as the complement of a) verbs expressing "directionality" (motion, giving/taking, sending), e. g. ***frumentatum*** *missa* fetch.grain-SUP-ACC[8] send-PFV.PASS.PTCP-ABL.SG.F 'sent to fetch grain' (PROIEL, `53469`),[9] in the accusative, and b) of adjectives of evaluation, e. g. *difficile* ***factu*** difficult-NOM.SG.N do-SUP-ABL 'difficult to do' (PROIEL, `86346`), in the ablative. Also recorded, even if marginally, but simply a subcase of the "active" use, is the periphrastic construction of the so-called "future"[10] passive infinitive, formed with the passive (with impersonal meaning) present infinitive *iri* from *eo* 'to go', as in e. g. *has tibi* ***redditum*** *iri [putabam]* this-ACC.PL.F you-DAT give.back-SUP-ACC go-IPFV.INF-PASS [think-IPFV.IND.PST-1SG] '[I thought] these would be returned to you' (PROIEL, `225189`). This supplies the supposed lack of a construction with the auxiliary *sum* 'to be' as in the very infrequent (cf. §6, Table 2) active equivalent *[eum] has tibi* ***redditurum*** *esse [putabam]* (constructed), using the so-called future participle instead.

The uses of the supine alternate with some infrequent constructions with the infinitive, such as the so-called infinitive of purpose as in *meridie* ***bibere*** *dato* noon-ABL.SG drink-IPFV.INF-ACT give-IMP.FUT-2SG 'give (them) to drink at noon' (Greenough et al., 2014, §460f.) (cf. the supine ***potum*** *dedi* drink-SUP-ACC give-PFV.IND.PRES-1SG 'I have given (you) to drink' PROIEL, `224782`), which might be reminescent of the origin of this verb form (Palmer, 1988, p. 319f.). In modern Romance languages the supine has been indeed replaced by the infinitive in all contexts, e. g. *ha mandato a* ***dire*** '(he/she/it) sent to tell (i. e. let know)' (Italian-VIT, `VIT-8312`; Alfieri and Tamburini (2016)) or *difficile da* ***raggiungere*** 'difficult to reach' (Italian-VIT, `VIT-242`),[11] and a direct descendant seems to survive only in Rumanian, as in *e* ***de mirat*** *cum trăieşte* 'it's amazing how he lives'[12] (Mallinson, 1988, §4).

The use of supine is already very sparse in the Latin data at our disposal: across all Latin UD treebanks, it occurs a mere 17 times[13] (5 times in the ablative), 16 of which are found in PROIEL, and it is so totally absent in corpora representing later varieties, but for one unusual case in UDante (`Mon-644`). However, morphologically identical abstract deverbative nouns of the 4th declension, annotated as `NOUN`s, are very common. In the UDante treebank (the only one where this information is available as of UD v2.9) more than 75% of fourth-declension noun (`NOUN`/`PROPN`) lemmas, i. e. 90 out of 120 (for a total of 421 out of 522 occurrences), are traceable back to supine forms,[14] e. g. *spiritus* 'spirit' from *spiro* 'to breath'. These

[5] In Latin grammars, the term "mood", beside indicative, subjunctive and imperative, often also encompasses nonfinite verb forms, even if these do not actually express a `Mood` in UD's sense.

[6] Quite uniquely, this grammar lists the supine under other participles, probably in the absence of a better choice.

[7] The excerpts from this grammar are presented here in the translation by the author.

[8] here and therafter, the gloss SUP stands for "supine"

[9] All quotations from UD corpora report the respective corpus code and sentence id (`sent_id`), while constructed examples are labelled as such. Forms under discussion are bold-faced, while arguments relevant to the discussion are underlined. Only for Latin samples, a linear gloss is given in the Leipzig formalism (see `https://www.eva.mpg.de/lingua/resources/glossing-rules.php`). A general reference for Latin is the classic grammar by Allen and Greenough (2014), or Barbieri's (1995).

[10] The aspectual notion of prospectivity would be probably more fitting here; cf. §6, Table 1.

[11] Translatable into Latin with a supine as *mittit* ***dictum*** and *difficile* ***peruentu*** (constructed).

[12] Where *de mirat* would correspond to Latin ablative supine *miratu*, or also dative *miratui*, from *miror* 'to marvel at'. We note that Rumanian treebanks do not use the value `Sup`, but seem to prefer `Part` or a treatment as `NOUN` instead.

[13] Tokens with UPOS `VERB` and `VerbForm=Sup`.

[14] Corresponding in Word Formation Latin (WFL), a resource for Latin derivational morphology (Litta and Passarotti, 2019),

nouns appear marked for every possible case and nominal dependency relation.[15] This observation could actually suggest, from a synchronic perspective,[16] to give purely nominal interpretations to the occurrences of supines: on the one hand, parallel to the "active" supine we have the accusative of direction (Greenough et al., 2014, §388b) as in *ire* ***Hyerosolimam*** go-IPFV.INF-ACT Jerusalem-ACC.SG 'go to Jerusalem' (PROIEL, `13700`), and on the other hand, parallel to the "passive" supine, the ablative of respect or specification, as in ***uirtute*** *praecedunt* virtue-ABL.SG excel-IPFV.IND.PRES-3PL 'they excel in courage' (Greenough et al., 2014, §418). While the latter interpretation seems justified, in the former case we can effectively find the supine taking arguments the same way as a corresponding finite predicate, e. g. *Tigranem ires* ***salutatum*** Tigranes-ACC.SG go-IPFV.SUB.PST-2SG greet-SUP-ACC '(so that) you would go to greet Tigranes' (PROIEL, `76590`), justifying its interpretation as a predicate, but then only limited to its accusative form.

The point of these observations is that the `VerbForm` value `Sup` appears, already on a language-internal point of view, not sufficiently focused: it seems to be based more on its etymological origin as a deverbative noun rather than on its synchronic function. But it would not be desirable to use `Sup` to represent a derivational process: derivational morphology is not really the focus of UD morpholexical features, and otherwise, for coherence, other deverbal nouns should also be marked for their derivation.[17] At the same time, it is reasonable to assume that the function of a verbal form like the crystallised accusative of the supine in Latin can be found in other languages, and so, in a crosslingustic perspective, another typological label would be better suited (cf. §6).

3.2 Supine in other languages and `Sup` in UD

The label of "supine" is variously used in grammars of languages other than Latin, but often the connection with the Latin supine does not appear fully motivated from a morphosyntactic point of view, a fact that corroborates the observations about the language-specificity of this label. As of UD v2.9, apart from Latin, 9 languages[18] make use of this feature; unfortunately, a documentation page is available for only 3 of them: Old Church Slavonic, Slovene and Swedish. Slovene and Swedish are also cited as examples in the universal guidelines, while the language-specific documentation for Old Church Slavonic and Slovenie is essentially identical, mirroring their close kinship. It follows an overview of these two supines, together with the Estonian one (the only non-Indo-European language together with North Sami) and their alleged relationship with the Latin supine:

Slovene (and some other current or old Slavic languages): it is an indeclinable verb form whose formation is linked to that of the infinitive (differing relatively to an *-i* suffix), apparently contrasting with it by expressing intentionality (Greenberg, 2006, §4.1.1.8), only appearing after verbs of motion "instead of infinitive" (UD guidelines) and capable of taking its own arguments: *grem domev* ***sežgat*** *dnevnik* 'I'm going home to burn (my) diary' (Priestly, 1993, §3.2.1, §4.5). If we ignore morphological differences, the communality on the syntactic level is thus partial, involving only the Latin accusative supine. We are not in the presence of a deverbative noun, but rather of a variant of the infinitive with lexically determined complementary distribution, so the `Inf` value might be a more fitting choice.[19]

mainly to rules 107 and 119 (conversions), 627 (suffix *it*), and 748 (suffix *at*). We notice that such figures seem to point out the fact that the protoypic Latin 4th-declension noun actually *is* a supine, as it were; however, a more thorough investigation over more extensive data is needed to support this claim.

[15] As a search for tokens satisfying `UPOS=NOUN`, `Gender=Masc`, `InflClass=IndEurU`, and, where present, `VerbForm=Sup`, in one of the Latin treebanks using these features can confirm.

[16] In Old Latin, contrarily, it is surely the case that deverbative nouns in general can take arguments like a corresponding finite predicate; cf. (Clackson and Horrocks, 2007, §4.2.3, c, iii).

[17] For example the extremely productive *(t)io(n)* suffix, as in *uisio* 'vision' from *uideo* 'to see', which according to WFL accounts for 2684 forms out of 14 418 recorded nouns (Litta and Passarotti, 2019).

[18] Estonian, Faroese, Icelandic, Marathi, North Sami, Old Church Slavonic, Old East Slavic, Slovene, Swedish.

[19] Diachronically, though, it is true that the Slavonic supine seems to be derived from the accusative case of a *u*-stem deverbative noun, too (Schenker, 1993, §3.2.2). This might be a further reason for its denomination.

Swedish : it is an indeclinable variant of the past (i. e. perfective) participle[20] used in the periphrastic construction of the composite past, based on the auxiliary verb *ha* 'to have', while in the passive construction (which has a predicative origin similarly as in Latin) the participle agrees in gender and number with the subject: *Jag har* ***ätit*** *maten* 'I have eaten dinner' vs. *Maten är äten* 'Dinner is eaten' (UD guidlines). As such, no connection whatsoever can be found with the Latin supine. Judging from the data, the notion of supine in Faroese and Icelandic is the same as in Swedish.

Estonian (not documented in UD): it essentially appears to be the inflected form of the infinitive under a different name, i. e. the infinitive fulfilling oblique roles, as opposed to the proper infinitive, used for core arguments (Viitso, 1998, p. 139). In this context, the infinitive in the illative case, representing motion to a place, comes closest to the "active" supine of Latin, as in *lähen malet* ***mängima*** 'I'm going (somewhere) to play chess'; but also *olin klubis malet* ***mängimas*** 'I was in the club playing chess' (inessive), with no parallels. The similarity with the Latin supine is again only syntactic and partial, and, representing a paradigmatic variation, like in Slovene it seems better captured by `Inf`, especially since it seems part of a full inflectional paradigm (with possible suppletive forms), and not defective like the Latin supine.

To sum it up, the uses of the `Sup` label outside Latin treebanks tendentially seem to rest upon vague parallels between the syntax of generic infinitival constructions and that of the Latin supine in the "active" construction (see §3.1): resemblances are however at best only partial (most often they do not include the "passive" usage of Latin), or are only part of a more extensive paradigm (like in Estonian). Beside that, none of these forms appear to be used in non-predicative constructions as is the equivalent Latin noun. Finally, we note that the generic status of "verbal noun" does not *per se* justify a preference for `Sup` with respect to other possible labels like `Inf` or `VNoun` (or even `Conv`), especially when these have a better appeal in the respective languages. As a comparison, it is interesting to notice that the grammatical tradition of Finnish, a language very closely related to Estonian, just uses the denomination "(third) infinitive" for the exact equivalent (including inflection, cf. Abondolo (1998)) of the Estonian "supine", as reflected by the use of the value `Inf` and absence of `Sup` in Finnish UD treebanks.

The real main reason for these traditional denominations thus seems to lie in the language-internal need to terminologically differentiate similar and correlated forms, e. g. infinitive in Slovene and Estonian, and participle in Swedish. Unfortunately, this brings along all the problems of excessive specificness already discussed for Latin in §3.1, and, on a typological level, is further misleading in that it establishes very specific, but not really grounded, parallels with the Latin form (the "original one", as it were), which are wanting also from a purely morphosyntactic point of view.

4 Distribution of nonfinite `VerbForm` values in UD

The picture that emerges from the discussion in §3.2 is that of a label, `Sup`, employed in UD not so much on the basis of morphosyntactic consideration, as for assonance, in deference to prior grammatical traditions; such traditions are themselves based on simultaneously superficial and too focused syntactic resemblances to phenomena originally studied for Latin, whose grammar, from ancient times, has long represented the "ideal grammar" in Europe.[21] With this, it is not meant that such distinctions are not useful or motivated internally to the given language but that, regrettably, these more or less successful attempts at following in the footsteps of Latin grammar terminology do not allow for meaningful interlinguistic comparisons, and often even contribute to the establishing of inaccurate analogies. A similar state of things appears also from the use of other nonfinite `VerbForm` values UD-wide, of which an overview follows (as of v2.8; where not specified, quotations are from universal guidelines) :

[20] Originally, its neuter singular form (Andersson, 1994, p. 284f.).

[21] While for a long time Latin grammarians themselves resorted to Greek grammar canon to frame Latin, see (Clackson and Horrocks, 2007, §6), also (Law, 2003, §4); see the pioneering work of Priscianus (Keil, 1855), and also cf. e. g. how the diverging phenomenon of *ablativus absolutus* was approached, as detailed in (Sluiter, 2000).

`Conv` employed by 36 languages for a verb form "which shares properties of verbs and adverbs", consequently appearing in an adverbial function, and so identified principally at a syntactic level (following the UD definition of `ADV` as "words that typically modify verbs for such categories as time, place, direction or manner"). Despite this value, some languages like Slovene and Latin opt to annotate possible candidates directly as `ADV`s (and consequent relation `advmod` instead of `advcl`), e. g. *sufficienter* suffice-IPFV.ACT.PTCP-ADV 'sufficiently'.[22] We note that the term "converb" first appeared in the field of Altaic studies (Haspelmath, 1995, §7) and has never been part of traditional terminologies of European languages.

`Gdv` employed by 4 languages (including Latin). The universal guidelines briefly state "used in Latin and Ancient Greek". While in Latin the gerundive is a kind of participle (see §6), the documentation for Armenian defines it as "a nonfinite verb form that shares properties of verbs and nouns", which would rather fit with `VNoun`. So, also this label appears to arise from traditional denominations, without being supported by a morphological definition; it is highly language-specific, and as such has not spread beyond these few languages (3 of which, Latin, Ancient Greek and Sanskrit, represent ancient phases of modern Indo-European languages, to which Armenian also belongs).

`Ger` employed by 21 languages (including Latin), even if deprecated by the universal guidelines. Here, the difference with `Conv` is not clear: e. g. the Italian gerund ***Arrivando*** *tardi si perde il treno* 'Arriving late you miss the train' (Italian guidelines) looks equivalent to Czech transgressive ***udělavši*** *večeři, zavolala rodinu ke stolu* 'having prepared the dinner, she called her family to the table' (universal guidelines), where it is labelled as `Conv`. Notably, while the latter inflects for gender, number and aspect, the former is invariable. The term derives from the Italian *gerundio* being the direct descendant of Latin *gerundium*, cf. again (Haspelmath, 1995, §7), itself a *gerundivum* (`Gdv`) in a particular syntactic context (see §6), of which it has kept the name despite radical morphosyntactic changes. In English treebanks, the use of `Ger` is contextual (but it is probbaly the case that we are dealing with two different homographic forms here): the same form is labelled as `Part` instead when preceded by the auxiliary verb *to be*; it is described (also universally) as "shar[ing] properties of verbs and nouns", which would rather lead to `VNoun`.

`Inf` employed by 75 languages (including Latin), it is together with `Part` the most universally used value, and at the same time the most undefined. Neither the universal guidelines nor any language-specific documentation put forward any true definition; the wide-spread identification as a citation form is of course purely conventional and extremely language-specific. Infinitive seems to be treated as a linguistic "primitive notion", self-evident for the languages it is applied to. However, the documentation for Irish, stating that "[t]he infinitive verb form is the same as the verbal noun", lets one question if `Inf` and `VNoun` are not actually referring to the same entity (see **`VNoun`** and cf. §6).

`Part` employed by 75 languages (including Latin), with a general agreement on it representing a verb form "shar[ing] properties of verbs and adjectives". This identification is bound to happen principally on a syntactic (but possibly also semantic) level, as for the UPOS `ADJ` itself, defined in the guidelines as "words that typically modify nouns and specify their properties or attributes". In some languages (as in Latin) morphological criteria might also be applied, but this is not a universal fact, and it is more often than not a consequence of a word being an adjective rather than the opposite. There can be contradictions, however: notwithstanding that the Latin gerundive shares morphology and syntax with other participial forms (see §6), the historical difference in naming convention (`Gdv` vs. `Part`) has been carried over into UD treebanks.

`VNoun` employed by 15 languages, it stands for "[v]erbal nouns other than infinitives"; however, being the value `Inf` undefined (see **`Inf`**), this leaves place for arbitrariety, and the distinction is not motivated. Indeed, despite the cases (cf. §3.2) in which this label could be appropriate, there seems to be

[22] Which is potentially accompanied by *suffecte* suffice-PFV.PASS.PTCP-ADV and *suffecture* suffice-PROSP.ACT.PTCP-ADV, thus showing a paradigmatical variation in aspect/voice, all from the same verb.

a general preference for Inf, probably influenced by Western naming traditions. Only 9 languages use both labels:[23] this complementarity goes into the direction of a factual equivalence of the two labels. Indeed, the Turkish documentation explicitly mentions this fact, claiming a preference for VNoun. In UD, NOUNs are defined as "a part of speech typically denoting a person, place, thing, animal or idea", pointing to mainly semantic criteria for their identification.

5 Identification of cardinal VerbForms in UD

From the overview in §4, an explicit distinction emerges between those values whose definition is oriented towards a specific UPOS with all respective morphosyntactic (and semantic) implications, i. e. Conv∼ADV, Part∼ADJ and VNoun∼NOUN, and the remaining ones (Gdv, Ger, Inf, Sup), whose definitions are left undetermined and/or which stem directly from traditional, language-specific denominations; no less than 3 (Gdv, Ger, Sup) originally refer to entities or constructions extremely peculiar to the Latin language, and have been adopted with various degrees of consistency by other grammatical traditions (cf. §3.2 and §4). Another issue is that such very specific labels isolate peculiar syntactic constructions which are not necessarily related to morphology, and which obscure the more general picture. The three UPOS-oriented values Conv, Part and VNoun can instead be seen as cardinal choices that logically reflect all possibilities contained in the morphosyntactic system of UD. They follow straight from the intuition that a verb form that keeps its lexicality continues to be head of a clause that can be itself embedded in a matrix clause in the same way as a non-verbal, i. e. nominal, phrase: so, in the UD formalism, either "mimicking" an adjective (ADJ), an adverb (ADV) or a noun (NOUN/PROPN), i. e. each and every lexical nominal part of speech in UD.

In the end, if the feature VerbForm stands to represent the morphosyntactic (and to some extent also semantic) properties that a verbal stem can assume in its inflectional paradigm (cf. §2), all the while keeping the possibility to act as the equivalent predicate of a main, independent clause with respect to its arguments, then, in agreement e. g. with Haspelmath (1995), we argue that a set of values mirroring all possible logically corresponding parts of speech in the given annotation formalism should suffice: in the case of UD, then, those which are conventionally labelled as Conv, Part and Inf/VNoun.[24] Consequently, the other labels are not actually needed and, on the contrary, do not contribute to the goal of inter-linguistic comparison implicit in the UD project, since they usually arise from idiosyncratic, language-specific terminology that conflicts with universal labels; we also argue that they can be all traced back to the three cardinal categories, or to specific (syntactic) behaviours of other deverbative parts of speech (not being truly part of a verbal paradigm in such a case, cf. §3.1). This will be done for Latin in the next section. Finally, we note that such reorganisation of verb forms around cardinal, UPOS-oriented values would not alter the extant possibility to assign a VerbForm to a non-VERB token, as such an assignment is of etymological rather than morpholexical character, and points to the paradigmatic origin of the form in question, not to its synchronic use; given the part-of-speech label, there subsists no ambiguity about this double connotation of the VerbForm feature.

6 Reorganising nonfinite Latin verb forms

As seen in §4, UD Latin treebanks currently make use of five out of seven values for nonfinite VerbForms, i. e. Gdv, Ger, Inf, Part and Sup. Their implementation is comparable between treebanks, as it more or less regularly follows traditional definitions. Below, Latin nonfinite forms are examined from a morphological and a syntactic point of view. Other considerations concerning when and whether some forms should be considered VERBs or else,[25] are out of the scope of this investigation.

[23] Erzya, Irish, Komi Zyrian, Mbya Guarani, Moksha, Polish, Skolt Sami, Turkish, Turkish German. But: Irish claims their identity (see **Inf**); Inf does not appear in the Turkish language-specific documentation; some of these languages share common annotation principles (e. g. Uralic languages, under the code urj; cf. Partanen et al. (2018, §3)). Thus, actual figures are lower.

[24] Noting that VNoun is a better choice, being less language-specific than Inf, and that Part should also be relabelled in

Denomination & **example *ago***	**VerbForm** **current**	**VerbForm** **proposed**	**Aspect**	**(Tense)**	**Voice**	**InflClass** **[nominal]**	**Gender**	**Number**	**Case**
Perfect participle *actus/a/um*	Part	Part	Perf	(Past)	Pass	IndEurA/ IndEurO	*	*	*
Present participle *agens*	Part	Part	Imp	(Pres)	Act	IndEurI	*	*	*
Future participle *acturus/a/um*	Part	Part	Prosp	(Fut)	Act	IndEurA/ IndEurO	*	*	*
Present infinitive *agere*, *agi*	Inf	VNoun	Imp	(Pres)	Act/Pass	Ind	*Neut*	*Sing*	*Acc/Nom*
Perfect infinitive *agisse*	Inf	VNoun	Perf	(Past)	Act	Ind	*Neut*	*Sing*	*Acc/Nom*
Gerund *agendo/um/o/i*	Ger	Part or VNoun	Prosp	(Fut/Pres)	Pass	IndEurO	Neut	Sing	Abl/Acc/ Dat/Gen
Gerundive *agendus/a/um*	Gdv	Part	Prosp	(Fut/Pres)	Pass	IndEurA/ IndEurO	*	*	*
Supine *actu/um*	Sup	Conv (or NOUN)	Prosp	–	Act	IndEurU	Masc	Sing	Abl/Acc

Table 1: Morphological properties of Latin nonfinite verb forms expressed in the UD formalism, with proposed VerbForm relabellings. Values for Tense are shown following their use in treebanks for legacy reasons only, since tense is not applicable to Latin nonfinite forms, cf. e. g. (Pinkster, 1990, §11.2.2). Legend: asterisk = all values possible; italics = inherent or contextual, not morphologically expressed values, i. e. not matched in the actual form (infinitives are indeclinable); hyphen = not observed; or = a different annotation might be possible for some contexts (see text). The example forms are limited to singular nominatives where possible, else all forms are listed. The value Voice is intended in a purely morphological, and not syntactic (clausal), sense.

Morphology Table 1 shows, in UD terms, the possible sets of values corresponding to the morpholexical features that are expressed by Latin nonfinite verb forms.[26] Notably, Mood, Tense and Person are absent, as in Latin they are a prerogative of so-called finite forms (see §2); Degree, being only optional, is also not shown. The split between two groups is evident: irrespective of different combinations of Aspect and Voice, one group (participles and gerundives) follows (prototypic) adjectives in not having an inherent, but only a relational[27] Gender/Number, inflecting for Case according to so-called 1st- ("*a* & *o* stems") and 2nd-class (specifically, "*i* stems") adjectival paradigms, and the possibility of being marked for Degree (e. g. *ardentiori* burn-IPFV.ACT.PTCP-CMPR-DAT.SG 'more burningly', UDante, Mon-283) ;[28] conversely, the other group (infinitives and supine) is similar to nouns, in that its members are bound to one given inflectional paradigm and/or possess a fixed, inherent gender and number, while case varies (even if defectively), and cannot express degree. Therefore, from this point of view, we have a natural partition into Part-forms and VNoun-forms, as discussed in §5. This means that both Ger and Gdv would be superseded by Part; morphologically, the identity of these two forms, specifically of the gerund as a particular case of gerundive, seems to be out of question (Haspelmath, 1987; Miller, 2000; Jasanoff, 2006). These choices are in fact already substantiated by traditional grammars: gerundive is considered a participle in (Greenough et al., 2014, §500), which "expresses the action of the verb in the form of an Adjective" (Greenough et al., 2014, §488), and is "a verbal adjective" according to

this sense.

[25] This problem becomes particularly relevant for later varieties of Latin. For example, cf. the treatment of *agens*, the present participle from *ago* 'to drive, to act', in the IT-TB (13th c. CE): either 'driving, acting', UPOS VERB (862 occurrences), or 'agent', UPOS NOUN (353 occurrences, including one incorrectly annotated as VERB). Conversely, no tokens with lemma *agens* are found in PROIEL.

[26] While these "schemata" are quite uncontroversial, the identification of a prospective aspect for some forms probably does not represent a common opinion; however, it is to be seen as the natural aspectual counterpart to the traditionally claimed (but inapplicable, see Table 1) future tense.

[27] That is, determined by agreement with another element, see relations in Table 2.

[28] They can also take an adverbial form, but are then regularly annotated as ADVs, not Convs, by all UD Latin treebanks; see §4, **Conv**.

(Barbieri, 1995, §164) ; gerund "is the neuter of the gerundive" (Greenough et al., 2014, §501); infinitive is "properly a noun" that "often admits the distinction of tense" (Greenough et al., 2014, §451), "a neuter singular verbal noun" (Barbieri, 1995, §151). However, the extreme defectiveness of the supine, which, as a predicate ("active" supine; the "passive" supine is to be treated as a simple `NOUN`, which is the standard for Latin deverbative nouns in Classical literature), appears only in the accusative case (cf. §3.1), sets it apart from more regular verbal nouns and instead supports a reading as a different `VerbForm`, namely a converb `Conv`: this analysys is further corroborated by the distribution of its syntactic relations.[29] In the same way, the uses of the `Gdv` identified as Ger can be interpreted as veering towards a less relational `VerbForm` than `Part`, and thus `VNoun`; more under **Syntax**.

`VerbForm` **current**	**`VerbForm`** **proposed**	**Denominations and respective frequencies in the data**	**Dependency relations with respective frequencies**
`Part`	`Part`	Perfect participle (37.88%)	**finite** **`acl`** **`advcl`** 33.67% 33.38% 18.52%
		Present participle (18.96%)	**`advcl`** **`acl`** **finite** **`root`** 36.39% 31.58% 9.86% 6.02%
		Future participle (0.59%)	**finite** **`ccomp`** **`root`** **`advcl`** 43.07% 35.10% 9.14% 5.01%
`Inf`	`VNoun`	Present infinitive (30.58%)	**`xcomp`** **`ccomp`** **`csubj`** 60.78% 18.35% 14.10%
		Perfect infinitive (1.14%)	**`ccomp`** **`xcomp`** **`root`** **`csubj`** 50.61% 21.93% 10.58% 8.59%
`Ger`	`Part` or `VNoun`	Gerund (6.73%)	**`advcl`** **`acl`** 49.01% 45.46%
`Gdv`	`Part`	Gerundive (4.08%)	**finite** **`root`** **`advcl`** **`acl`** **`ccomp`** 37.31% 19.17% 14.55% 10.23% 10.01%
`Sup`	`Conv` (or `NOUN`)	Supine (0.03%)	**`advcl`** **finite** 94.12% 5.88%

Table 2: Distribution of nonfinite verb forms and their most frequent (≥5%) dependency relations in UD Latin treebanks, broken down by traditional denominations. Only tokens with UPOS `VERB`, i. e. deemed to have the same argument structure as the predicate of a main, independent clause, and with `VerbForm` different than `Fin`, have been taken into consideration, for a total of 57 411 tokens. Relation subtypes (e. g. `advcl:pred` w. r. t. `advcl`) have been neutralised to compensate for different annotation styles. The underspecified relation `conj` (6439 occurrences) has been traced back and substituted with the relation of the respective co-ordination "head". The label **finite** comprises all nonfinite forms which have a dependent node labelled with `aux` or `cop` (9510 occurrences): for all purposes, these combinations are or derive from finite, albeit periphrastic, predicates, and so their exact syntactic relations are no longer relevant in this context. Annotation errors and inconsistencies, together with elliptic clauses, produce noise in the figures: e. g., `root` and `ccomp` labels are in many cases clues for elliptic, periphrastic, finite predicates, as e. g. in the formula ***dicendum*** *~~est~~ quod…* 'it is to / will be said that…', where *est* is the auxiliary 'to be' (IT-TB) and *dicendum* a gerundive.

Syntax Table 2 summarises the distribution of syntactic roles, as per UD dependency relations, in all available Latin treebanks.[30] As expected, nominal relations are negligible, and syntactic data appear to

[29] We would like to thank an anonymous reviewer for pointing out this fact, which is *a posteriori* self-evident: another example of how traditional, entrenched points of view ("Latin has no converbs") often stand in the way of typological awareness.

[30] We note that this overview necessarily represents a mean of diachronic, diastratic and diaphasic varieties (cf. §1), but because of the increasing status, since late antiquity (ca 4th-5th c. CE), of Latin as an international and prestigious *lingua franca* rather than a living and native language, see (Clackson and Horrocks, 2007, §8), also (Wright, 1998; Leonhardt, 2009), we

be in nearly perfect agreement with morphology in Latin. So we observe that, on the one hand, infinitive forms are specialised as heads of clauses that fulfil core arguments, which are prototypically occupied by `NOUN`s (given the parallels `xcomp`/`ccomp`∼`obj`, `csubj`∼`nsubj`), and so `VNoun` becomes the natural choice here, as discussed in §4 and §5. On the other hand, the gerundive has the same profile as participial forms: it resembles the most the future participle (also for being rather infrequent), which might not be a coincidence considering the common prospective aspect, which appears to have been marginalised in favour of the main imperfective/perfective opposition.[31] If the latter is a `Part`, then so is the gerundive; the different functional distributions of participles might be possibly explained in semantic terms tied to aspect and other features, and could mirror preferences with regard to which elements can appear as so-called secondary predicates (Pinkster, 1990, §8.3), but overall they are seen to fulfil attributive/predicative roles (cf. "dominant participles" Pinkster (1990, §7.4.7)). On the contrary, the supine, appearing nearly exclusively in an adverbial function, cannot itself agree with any subject as participles do (cf. Nikitina and Haug (2016)) and e. g. cannot appear in absolute constructions,[32] confirming its status as `Conv`[33] (or `NOUN`), as seen for its effectively absent inflectional morphology (compared e. g. to the complete paradigm of a participle). Finally, the syntax of the gerund is more difficult to assess on a generic level: diachronic and diaphasic distinctions are needed. In fact, while the gerund is clearly an inflected form of the gerundive (Haspelmath, 1987; Miller, 2000; Jasanoff, 2006) and so supposedly passive, it is considered distinct from it on the syntactic ground that it can govern an object instead of a subject (without agreeing with it). For example, we find the adnominal (`acl`) *[necessitas] plura nomina deo* ***dandi*** [necessity-NOM.SG[.F]] more-ACC.PL.N noun-ACC.PL[.N] god-DAT.SG[.M] give-PROSP.PASS.PTCP-GEN.SG.N '[the need] of giving God more nouns' (IT-TB, `train-s1483`), with direct object in the accusative case and uncontroversially transitive reading, in place of a Classically expected passive construction *plurium nominum* ***dandorum*** more-GEN.PL.N noun-GEN.PL[.N] give-PROSP.PASS.PTCP-GEN.PL.N, lit. 'for more names to be given', in the genitive (notice that this is a not head-coreferent clause embedded as an adnominal modifier, so there is no agreement with *necessitas*). In the same text (the *Summa contra gentiles*), we also find *[necessitatem]* ***sustentandi*** *corporis* [necessity-ACC.SG[.F]] sustain-PROSP.PASS.PTCP-GEN.SG.N body-GEN.SG[.N], lit. '[the need] of the body being sustained', i. e. 'to sustain the body' (IT-TB, `train-s22169`). Thus, we agree with Haspelmath (1987, §5.2) that, especially when such an alternation can still be found in a significative ratio as in the IT-TB (565 vs. 509 occurrences respectively), the gerund can be simply explained in terms of a gerundive with impersonal value and deponent[34] (Greenough et al., 2014, §190) behaviour, and so, in the annotation, we can trace it back to `Part`. Further, of the 1459 identical[35] nonfinite clauses headed by a `Ger` in the UD Latin treebanks, at most 635 have a clear direct object: this means that for more than half of `Ger`-clause types a plainer interpretation as gerundives (`Part`) is also possible, and so preferable in general. But in some contexts, the at first only occasional (Miller, 2000) reanalysis from a passive to

can regard linguistic change in written sources as extremely moderate, "frozen" by the adherence to the prestigious Classical standards, in comparison to the contemporary processes that lead to modern Romance languages (Väänänen, 1981; Palmer, 1988; Ledgeway, 2012), which were gradual anyway, cf.Wüest (1998). So, an aggregated picture keeps its significance here.

[31] In fact, both forms (together with the supine) have disappeared in modern Romance languages, together with a morphologically expressed inchoative aspect, leaving only fossilised lexemes, see (Harris, 1988, §3). An explanation for this might be that prospective adjectival/adverbial forms are less time-stable than prototypical adjectives/adverbs, and so are preferentially expressed by "finite" predicates by languages, cf. (Stassen, 2003, §5), eventually leading, in the case of Latin, to their exclusion.

[32] An absolute construction is a nonfinite embedded adverbial clause with a subject different from any actors of its matrix clause, and, at least for Latin, headed by a paticipial form which agrees in gender and the number with its subject, both in the ablative case.

[33] Specifically, of purpose, with same or main subject than its matrix clause.

[34] Deponency can be seen, in general terms, as a mismatch between canonical morphological and syntactical behaviours (Baerman, 2007): in Latin, this happens for verbs displaying a passive morphology, but a transitive/active syntactical behaviour, e. g. *sequor* 'to follow (someone)', receiving a direct object argument in the accusative case. For a brief sketch of the problem posed by Latin deponent verbs, cf. discussion at `https://github.com/UniversalDependencies/docs/issues/713`.

[35] With the same forms in the same order, considering only the head and all possible core or functional dependent nodes (`obj`, `ccomp`, `xcomp`, `nsubj`, `csubj`, `mark`). So, *ad censum uobis* ***perexoluendum***, *ad censum nobis* ***perexoluendum*** and *ad censum* ***perexoluendum*** 'for tributes to be quitted [by you/us]' (LLCT) are all considered the same clause. This equivalence is needed to deal with formulaic repetitions especially in LLCT, where e. g. *ad censum perexoluendum* alone is repeated 68 times, or *(ad) legem et iustitia[m] faciendum* 'to carry out law and justice' 134 times.

an active construction of the spoken language may appear stabilised also in written documents: so, in the LLCT treebank, representing a Latin heavily influenced by early medieval Romance varieties, active gerunds like *[potestas]* ***remittendi*** *peccata* [power-NOM.SG[.F]] put.back-PROSP.PASS.PTCP-GEN.SG.N sin-ACC.PL[.N] '[the power] to forgive sins' (LLCT, `train-s21623`) (instead of an expected ***remittendorum*** *peccatorum* put.back-PROSP.PASS.PTCP-GEN.PL.N sin-GEN.PL[.N], in the genitive) represent 649 (97.89%) occurrences among a total of 663 `Gdv/Ger` constructions with core arguments, whereas only 14 clear cases of Classical gerundival constructions like in *ad* ***dedicandam*** *ipsam bassilicam* to dedicate-PROSP.PASS.PTCP-ACC.SG.F same-ACC.SG.F basilica-ACC.SG[.F] 'for this basilica to be consecrated' (LLCT, `train-s835`) can be found. Of the former, 63 occurrences, like the 40 variations of *non pondum* ***leuandum*** not weight-ACC/NOM.SG[.N] lift-PROSP.PASS.PTCP-ACC/NOM.SG.N '(the) taking away not a (single) pound' have an ambiguous reading, since the alleged object is a neuter singular like the gerund. The situation is reversed in PROIEL, more skewed towards Classical latin: 49 (13.46%) gerunds with object and 315 (86.54%) gerundives with passive subject (out of 364 `Gdv/Ger` with core arguments).

It so appears that the Latin nonfinite verbal system is naturally and, in a crosslinguistic prospective, effectively explained and annotated in terms of the only three labels `Part`, `VNoun` and `Conv`, avoiding the too language-specific and idiosyncratic values `Gdv`, `Ger` and `Sup`, and substituing `Inf` with a more universal label. Even when using identical `VerbForm` values, all forms identified by traditional grammars are kept distinct by virtue of morphological features or syntactic dependencies; conversely, were two forms not to be distinguished at any level (like the object-, subjectless gerund), the reasons for keeping them distinct would become questionable.[36] Finally, the Latin system is, despite what could show through traditional grammar, seen to possess a `Conv` form, whose presence is however marginal already in Classical times and completely outshined by the use of so-called "conjoined participles" (*participia coniuncta*) and/or eminently participial/adjectival absolute constructions (*ablativus absolutus* and secondary predications) with similar adverbial functions. Only much later, in Romance varieties, a crystallised gerund takes on the form of a converb.

7 Conclusion and last remarks

This paper proposes a reorganisation of the annotation of nonfinite Latin verb forms in the UD formalism (§3.1, §6), accomplished with respect to the morpholexical feature `VerbForm`, situating it in the wider perspective of achieving a simpler and "more universal", crosslinguistically valid system than the current one (§2, §5), highlighting the inconsistencies in its implementation across UD treebanks, also by Latin treebanks themselves (§3.2, §4). Latin has been chosen as a testbed, beyond showing extensive nonfinite verb formations and falling into the competences of the author (contributor to the IT-TB, LLCT and UDante Latin trebanks), especially because of its particular position at the origin, more or less foundedly, of a large part of (traditional) grammatical terminology, notably of European languages, which is encountered again in UD (e. g. `Inf`, `Part`, etc. for the feature `VerbForm`). Contributions from the work on typologically radically different languages would be a highly valued complement to the survey in §4, and to spark discussion about this topic in the UD community is one of the major goals of this paper.

We can lastly briefly mention two possible additions to the system, left for future examination, as a corollary to the discussion in §5: a) the introduction of a fourth nonfinite `VerbForm` value for highly specialised, frozen forms like the Swedish supine (§3.2), with a probable orientation towards an `AUX`-like category; b) the introduction of a "terminological feature"[37] that, parallelly to the constellation of UD morphosyntactic features/UPOS/relations characterising a verb form, would help retrieve it through its traditional denomination, thereby acknowledging historical, common language-specific conventions without however interfering with the universal analysis.

[36] And this seems to be the case for the Slovene infinitive and supine, who might be seen as the same form in a lexically determined, complementary allomorphic variation: then, the reading of intentionality (§3.2) would actually depend on the predicate rather than on the form itself.

[37] Cf. discussion at `https://github.com/UniversalDependencies/docs/issues/775`.

References

Daniel Abondolo. 1998. Finnish. In Daniel Abondolo, editor, *The Uralic Languages*, Language family descriptions, pages 115–148, London, UK; New York, NY, USA. Routledge.

Linda Alfieri and Fabio Tamburini. 2016. (Almost) Automatic Conversion of the Venice Italian Treebank into the Merged Italian Dependency Treebank Format. In *Third Italian Conference on Computational Linguistics - CLiC-IT 2016*, pages 19–23, Naples, Italy, December. CEUR-WS.org.

Erik Andersson. 1994. Swedish. In Ekkehard König and Johan van der Auwera, editors, *The Germanic Languages*, Language family descriptions, pages 271–312, London, UK; New York, NY, USA. Routledge.

Matthew Baerman. 2007. Morphological Typology of Deponency. In Matthew Baerman, Greville G. Corbett, Dunstan Brown, and Andrew Hippisley, editors, *Deponency and Morphological Mismatches*, number 145 in Proceedings of the British Academy, Oxford, UK. British Academy (through Oxford University Press).

David Bamman and Gregory Crane. 2011. The ancient Greek and Latin dependency treebanks. In Caroline Sporleder, Antal van den Bosch, and Kalliopi Zervanou, editors, *Language Technology for Cultural Heritage*, Theory and Applications of Natural Language Processing, pages 79–98, Berlin/Heidelberg, Germany. Springer.

Giovanna Barbieri. 1995. *Nuovo corso di lingua latina*. Loescher, Turin, Italy.

Flavio Massimiliano Cecchini, Marco Passarotti, Paola Marongiu, and Daniel Zeman. 2018. Challenges in Converting the Index Thomisticus Treebank into Universal Dependencies. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 27–36, Brussels, Belgium, November. Association for Computational Linguistics.

Flavio Massimiliano Cecchini, Timo Korkiakangas, and Marco Passarotti. 2020a. A New Latin Treebank for Universal Dependencies: Charters between Ancient Latin and Romance Languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 933–942, Marseille, France, May. European Language Resources Association.

Flavio Massimiliano Cecchini, Rachele Sprugnoli, Giovanni Moretti, and Marco Passarotti. 2020b. UDante: First Steps Towards the Universal Dependencies Treebank of Dante's Latin Works. In *Seventh Italian Conference on Computational Linguistics - CLiC-IT 2020*, pages 1–7, Bologna, Italy, March. CEUR-WS.org.

James Clackson and Geoffrey Horrocks. 2007. *The Blackwell History of the Latin Language*. Blackwell Publishing, Malden, MA, USA.

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

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

Hanne Eckhoff, Kristin Bech, Gerlof Bouma, Kristine Eide, Dag Haug, Odd Einar Haugen, and Marius Jøhndal. 2018. The PROIEL treebank family: a standard for early attestations of Indo-European languages. *Language Resources and Evaluation*, 52(1):29–65.

Marc L. Greenberg. 2006. *A Short Reference Grammar of Standard Slovene*. University of Kansas. Available online at `http://www.seelrc.org:8080/grammar/pdf/stand_alone_slovene.pdf`.

J. B. Greenough, G. L. Kittredge, A. A. Howard, and Benjamin L. D'Ooge. 2014. *New Latin Grammar for Schools and Colleges*, volume 1. Dickinson College Commentaries, Carlisle, PA, USA. Available online at `https://dcc.dickinson.edu/grammar/latin/credits-and-reuse`.

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, April. Association for Computational Linguistics.

Martin Harris. 1988. The Romance Languages. In Martin Harris and Nigel Vincent, editors, *The Romance languages*, Language family descriptions, pages 1–25, London, UK; New York, NY, USA. Routledge.

Martin Haspelmath. 1987. Verbal noun or verbal adjective? The case of the Latin gerundive and gerund. Arbeitspapier N. F. 3, Institut für Sprachwissenschaft, Universität Köln. Available online at `http://publikationen.ub.uni-frankfurt.de/frontdoor/index/index/docId/24318`.

Martin Haspelmath. 1995. Converbs in cross-linguistic perspective. Number 13 in Empirical Approaches to Language Typology, pages 1–55, Berlin, Germany. Mouton de Gruyter.

Martin Haspelmath. 2009. Terminology of case. In Andrej Malchukov and Andrew Spencer, editors, *Handbook of Case*, pages 505–517, Oxford, UK. Oxford University Press.

Jay H. Jasanoff. 2006. The Origin of the Latin Gerund and Gerundive: A New Proposal. *Harvard Ukrainian Studies*, 28(1/4):195–208.

Heinrich Keil. 1855. Prisciani Institutionum grammaticarum libri XVIII (ex recensione Martini Hertzii). In *Grammatici latini*, volume 3, Leipzig, Germany. Teubner.

Maria Koptjevskaja-Tamm. 1994. Finiteness. In Ronald E. Asher and J. M. Y. Simpson, editors, *The Encyclopedia of Language and Linguistics*, volume 3, pages 1245–1248, Oxford, UK. Pergamon Press. NB: This entry does not appear in the second edition of the *Encyclopedia*.

Maria Koptjevskaja-Tamm. 2006. Nominalization. In Keith Brown, editor, *Encyclopedia of Language and Linguistics*, volume 8, pages 652–659, Amsterdam, Netherlands. Elsevier. Second Edition.

Timo Korkiakangas and Marco Passarotti. 2011. Challenges in Annotating Medieval Latin Charters. *Journal for Language Technology and Computational Linguistics*, 26(2):103–114.

Vivien Law. 2003. *The History of Linguistics in Europe*. Cambridge University Press, Cambridge, UK.

Adam Ledgeway. 2012. *From Latin to Romance*. Number 1 in Oxford studies in historical and diachronic linguistics. Oxford University Press, Oxford, UK.

Jürgen Leonhardt. 2009. *Latein. Geschichte einer Weltsprache*. Beck, Munich, Germany.

Eleonora Litta and Marco Passarotti. 2019. (When) inflection needs derivation: a word formation lexicon for Latin. In Nigel Holmes, Marijke Ottink, Josine Schrickx, and Maria Selig, editors, *Words and Sounds*, pages 224–239. De Gruyter, Berlin, Germany; Boston, MA, USA, December. Interrogable online at http://wfl.marginalia.it/.

Graham Mallinson. 1988. Rumanian. In Martin Harris and Nigel Vincent, editors, *The Romance languages*, Language family descriptions, pages 391–419, London, UK; New York, NY, USA. Routledge.

D. Gary Miller. 2000. Gerund and gerundive in Latin. *Diachronica*, 17(2):293–349.

Tatiana Nikitina and Dag Trygve Truslew Haug. 2016. Syntactic nominalization in Latin: a case of non-canonical agreement. *Transactions of the Philological Society*, 25–50(1):195–208.

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 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May. European Language Resources Association.

Leonard Robert Palmer. 1988. *The Latin Language*. University of Oklahoma Press, Norman, OK, USA.

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

Marco Passarotti. 2019. The Project of the Index Thomisticus Treebank. In Monica Berti, editor, *Digital Classical Philology: Ancient Greek and Latin in the Digital Revolution*, number 10 in Age of Access? Grundfragen der Informationsgesellschaft, Berlin, Germany; Boston, MA, USA. De Gruyter Saur.

Harm Pinkster. 1990. *Latin Syntax and Semantics*. Romance linguistics. Routledge, London, UK; New York, NY, USA.

T. M. S. Priestly. 1993. Slovene. In Bernard Comrie and Greville G. Corbett, editors, *The Slavonic Languages*, Language family descriptions, pages 388–451, London, UK; New York, NY, USA. Routledge.

Alexander M. Schenker. 1993. Proto-slavonic. In Bernard Comrie and Greville G. Corbett, editors, *The Slavonic Languages*, Language family descriptions, pages 60–121, London, UK; New York, NY, USA. Routledge.

Anna Siewierska. 2013. Verbal person marking. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig, Germany. https://wals.info/chapter/102.

Ineke Sluiter. 2000. Seven grammarians on the ablative absolute. *Historiographia Linguistica*, XXVII(2/3):379–414.

Leon Stassen. 2003. *Intransitive Predication*. Oxford Studies in Typology and Linguistic Theory. Oxford University Press, Oxford, UK.

Mirko Tavoni. 2011. DanteSearch: il corpus delle opere volgari e latine di Dante lemmatizzate con marcatura grammaticale e sintattica. In Anna Cerbo, Roberto Mondola, Aleksandra Žabjek, and Ciro Di Fiore, editors, *Lectura Dantis 2002-2009. Omaggio a Vincenzo Placella per i suoi settanta anni*, volume 2 (2004–2005), pages 583–608, Naples, Italy. Il Torcoliere – Officine Grafico-Editoriali di Ateneo.

Veikko Väänänen. 1981. *Introduction au latin vulgaire*. Klincksieck, Paris, France.

Tiit-Rein Viitso. 1998. Estonian. In Daniel Abondolo, editor, *The Uralic Languages*, Language family descriptions, pages 149–183, London, UK; New York, NY, USA. Routledge.

Françoise Waquet. 2001. *Latin or the Empire of a Sign*. Verso, Brooklyn, NY, USA; London, UK. Translation of: Françoise Waquet. 1998. *Le latin ou l'empire d'un signe*. L'évolution de l'humanité. Albin Michel, Paris, France.

Roger Wright. 1998. Il latino: da madrelingua nativa a lingua straniera. In *La transizione dal latino alle lingue romanze. Atti della tavola rotonda di linguistica storica, Università Ca' Foscari di Venezia, 14–15 giugno 1996*, pages 77–85, Tübingen, Germany. Niemayer.

Jakob Wüest. 1998. Pour une linguistique historique non linéaire: Les formes analytiques du latin. In *La transizione dal latino alle lingue romanze. Atti della tavola rotonda di linguistica storica, Università Ca' Foscari di Venezia, 14–15 giugno 1996*, pages 87–98, Tübingen, Germany. Niemayer.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, Thórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Deniz Baran Aslan, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkadhur Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Mihaela Cristescu, Philemon Daniel, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phu'o'ng Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, Lorena Martín-Rodríguez, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Lu’o’ng Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adéday`ọ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Dmitry Sichinava, Janine Siewert, Einar Freyr Sigurdhsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Shafi Sourov, Carolyn Spadine, Rachele Sprugnoli, Steinthór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2021. Universal dependencies 2.9. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, http://hdl.handle.net/11234/1-4611. Licence Universal Dependencies v2.9.

Towards Universal Dependencies for Bribri

Rolando Coto-Solano[1] **Sharid Loáiciga**[2] **Sofía Flores-Solórzano**[3]
[1]Department of Linguistics, Dartmouth College
[2]CLASP, Department of Philosophy, Linguistics & Theory of Science, University of Gothenburg
[3]Ministry of Public Education, Costa Rica
rolando.a.coto.solano@dartmouth.edu sharid.loaiciga@gu.se
sofia.flores.solorzano@mep.go.cr

Abstract

This paper presents a first attempt to apply Universal Dependencies (Nivre et al., 2016; de Marneffe et al., 2021) to Bribri, an Indigenous language from Costa Rica belonging to the Chibchan family. There is limited previous work on Bribri NLP, so we also present a proposal for a dependency parser, as well as a listing of structures that were challenging to parse (e.g. flexible word order, verbal sequences, arguments of intransitive verbs and mismatches between the tense systems of Bribri and UD). We also list some of the challenges in performing NLP with an extremely low-resource Indigenous language, including issues with tokenization, data normalization and the training of tools like POS taggers which are necessary for the parsing. In total we collected 150 sentences (760 words) from publicly available sources like grammar books and corpora. We then used a context-free grammar for the initial parse, and then applied the head-floating algorithm in Xia and Palmer (2001) to automatically generate dependency parses. This work is a first step towards building a UD treebank for Bribri, and we hope to use this tool to improve the documentation of the language and develop language-learning materials and NLP tools like chatbots and question answering-systems.

Resumen

Este artículo presenta un primer intento de aplicar Dependencias Universales (Nivre et al., 2016; de Marneffe et al., 2021) al bribri, una lengua indígena chibchense de Costa Rica. Dado el limitado trabajo existente en procesamiento de lenguaje natural (PLN) en bribri incluimos también una propuesta para un analizador sintáctico de dependencias, así como una lista de estructuras difíciles de analizar (e.g. palabras con orden flexible, secuencias verbales, argumentos de verbos intransitivos y diferencias entre el sistema verbal del bribri y los rasgos morfológicos de UD). También mencionamos algunos retos del PLN en lenguas indígenas extremadamente bajas en recursos, como la tokenización, la normalización de los datos y el entrenamiento de herramientas como el etiquetado gramatical, necesario para el análisis sintáctico. Se recolectaron 150 oraciones (760 palabras) de fuentes públicas como gramáticas y corpus y se usó una gramática libre de contexto para el análisis inicial. Luego se aplicó el algoritmo de flotación de cabezas de Xia y Palmer (2001) para generar automáticamente los análisis sintácticos de dependencias. Este es el primer paso hacia la construcción de un treebank de dependencias en bribri. Esperamos usar esta herramienta para mejorar la documentación de la lengua y desarrollar materiales de aprendizaje de la lengua y herramientas de PLN como chatbots y sistemas de pregunta-respuesta.

1 Introduction

This paper presents a first attempt to conduct dependency parsing in Bribri, an Indigenous language spoken in southern Costa Rica (Glottolog `brib1243`). There is an increasing number of Universal Dependency treebanks (Nivre et al., 2016; de Marneffe et al., 2021) available for Indigenous languages of the Americas: e.g. Yupik (Chen et al., 2020; Park et al., 2021), Arapaho (Wagner et al., 2016), Hupa (Spence et al., 2018), Maya K'iche' (Tyers and Henderson, 2021), Shipibo-Konibo (Vasquez et al., 2018), Guaraní (Thomas, 2019), Apurinã (Rueter et al., 2021) and several Tupí languages from Brazil (Ferraz Gerardi et al., 2021).[1] However, there is no previous work on any member of the Chibchan family, a language family spoken in lower Central America, Colombia and Venezuela, so this paper seeks to address this gap and contribute to the automated syntactic analysis of these languages.

Bribri is a Chibchan language spoken by approximately 7000 people (INEC, 2011). It is a vulnerable language (Moseley, 2010; Sánchez Avendaño, 2013), still spoken by many adults and some children but mostly restricted to settings inside the home. Bribri is an morphologically ergative language (McGregor, 2009; Quesada, 1999; Pacchiarotti and Kulikov, 2021), with SOV word ordering, head-internal relative clauses and numeral classifiers. There has been some previous work on Bribri NLP: The first was the keyboard of Flores-Solórzano (2010), which allowed the language to be typed easily into computers and cellphones. The language also has an electronic Bribri-Spanish dictionary (Krohn, 2020; Krohn, 2021) and a morphological analyzer (Flores-Solórzano, 2019; Flores-Solórzano, 2017b), and there have been experiments in speech recognition (Coto-Solano, 2021), forced alignment (Coto-Solano and Flores-Solórzano, 2016; Coto-Solano and Flores-Solórzano, 2017), neural machine translation (Feldman and Coto-Solano, 2020; Mager et al., 2021) and natural language inference (Ebrahimi et al., 2021). This paper seeks to expand the work of Bribri NLP into the area of syntax and automated parsing, in the hopes of generating tools that help in the documentation and ultimately the revitalization of the language.

2 Methodology

In this section we will present the workflow that we followed for this first experiment. We collected sentences from various data sources (grammar books and oral corpora). We then tokenized the sentences and extracted the POS tag for each word. After that we designed a constituent grammar to perform the first automatic parse, and an algorithm to convert those constituent parses into dependency parses.

2.1 Data sources

For this first attempt we selected 150 sentences, containing 760 words. These ranged in complexity from simple structures (e.g. *Shkèna̱* 'Hello', lit. 'to have woken up') to entire conversations. For example, the longest sentence comes from an oral narration and contains 58 words. The sentences come from either published or Creative Commons licensed sources, specifically the textbook of Constenla et al. (2004), the grammar of Jara (2018) and the spoken Bribri corpus of Flores-Solórzano (2017a), and they included examples from the Amubri, Coroma and Salitre dialects. Most sentences were isolated examples, originally intended to illustrate the grammar of Bribri and chosen for the variety of their syntantic structures. However, the dataset also includes two short stories; one of them is in conversational style and it includes speech phenomena such as *reparanda* disfluencies (Universal Dependencies Contributors, 2021).

One major challenge is the normalization of the written data. As is the case with many Indigenous languages, where the orthography is of recent creation and created by outsiders to the community, there is considerable variation in how Bribri is represented in writing. There are four main sources of variation: (a) Different authors use different writing systems. For example, Constenla et al. (2004) uses a line underneath the vowel to indicate nasality, whereas Jara (2018) uses a tilde diacritic and Margery (2005) uses a Polish hook. Therefore, the word 'pot' can be found as *ù̱*, *ũ̀* or *ų̀* (all of them pronounced [ũ˥˥])[2]. (b) Phonological variation is not represented consistently. For example, the word *a̱mì̱* [ã˧ˈmĩ˥˥] 'mother' can also be written *mì̱* because the unstressed vowel in the first syllable can be deleted. (c) There is variation across dialects. The word 'road', for example, is *ña̱là̱* [ɲã˧ˈɽã˥˥] in the Amubri dialect and *ñolõ̀* [ɲõ˧ˈɽõ˥˥] in the Coroma dialect. Finally, (d) there is considerable idiosyncratic variation in and between documents, as would be expected of any language where the writing system has been recently adopted. During this work, the word 'much' has been found as *ta̱î̱* (Constenla et al., 2004), *tãì*, *tãı̃̀*, *tãı̃́* (Jara, 2018), *tã̂ĩ* (Pacchiarotti and Kulikov, 2021), *ta̱i̱*, *tá̱i̱*, *ta̱í̱*, *tái*, *taí*, *tá̱in*, *táin*, *taín* and *táìn* (MEP, 2017).

[1] There are some non-UD treebanks for languages like Quechua (Rios et al., 2008) and Karuk (Garrett et al., 2013).

[2] Bribri is tonal: The high tone is indicated by an grave diacritic (ù), the falling tone by the acute diacritic (ú), the low rising tone by a circumflex (û), and the low/neutral tone (Coto-Solano, 2015) is indicated by the lack of a diacritic (u).

The two NLP tools publicly available for Bribri, the keyboard layouts and the morphological analyzer (section 2.2 below) use the Constenla orthography. It is also used by the Ministry of Education of Costa Rica in school classes. Therefore, we will use that system here. However, when the Bribri treebanks are released, they will be made available in the two main orthographies, the ones in Constenla et al. (2004) and Jara (2018), and some orthographic variation might have to be standardized. When the automated dependency parser is released, it will have to be made resilient to the variation exemplified above, so that it can effectively tag and parse text that deviates from spelling norms. This is particularly important because, given Bribri's status as a vulnerable language, the main role of researchers at this stage should be to incentivize the creation of Bribri written materials, not to strictly enforce orthographic standards.

2.2 Tokenization and POS Tagging

The oral corpus in Bribri.net (Flores-Solórzano, 2017a) includes a unigram-based morphological analyzer (Flores-Solórzano, 2019). This program uses the finite-state analyzer FOMA (Hulden, 2009) to analyze each word. Example (1) shows Bribri words and their FOMA output. The FOMA was then used to extract the lemma and to extrapolate the part-of-speech for each word. For example, the word *ù* 'house' has the FOMA `ù+Sust` 'noun', so this word would be tagged as a noun with the lemma `ù`.

(1)

Bribri	Ye'	tö	ù	sú̱
FOMA	`+1PSg`	`+Posp[Erg]`	`ù+Sust`	`su̱+V+PerfImp`
Gloss	*I*	ERG	*house*	*saw*

'I saw the house' (Constenla et al., 2004, 52)

Because the program was unigram based, it is not sensitive to context and its output can include several possibilities for the morphological analysis. For example, the word *tö* is the ergative marker in sentence (1). When this word is entered as input to the FOMA, it produces three different outputs. These were used in combination with the surrounding words to decide the most appropriate POS for a given word.

One important issue for future work is tokenization. There are a few forms, like the reduced ergative marker *r* and the clitic pronouns, that can be attached to other words. The examples in (2) show the 3rd person absolutive clitic. Different authors deal with the clitic in different ways: they attach it directly to the verb, as in (2a), they separate it with a dash, as in (2b), or sometimes they write it separately, as in (2c). In this first experiment the clitics and the ergative markers were separated manually and stored separate from other words, but the parser needs to be made more resilient to these variations.

(2) a.

E'ku̱é̱k	és	**i**kíe	dör
because	like.this	**3SG.ABS**=to.be.called	COP

'That's why they call it like this' (García Segura, 2016, 11)

b.

Ie'	mì̱ne̱	**i**-ma̱u̱k
3SG	went	**3SG.ABS**=tie.INF

'She went to tie it up' (Constenla et al., 2004, 47)

c.

Ma̱	se'	tö	**i**	kiè	e̱ma̱	dláshawö
well	we	ERG	**3SG.ABS**	call	well	ginger

'Well, we call it, ah, ginger' (Valengana and Flores-Solórzano, 2017)

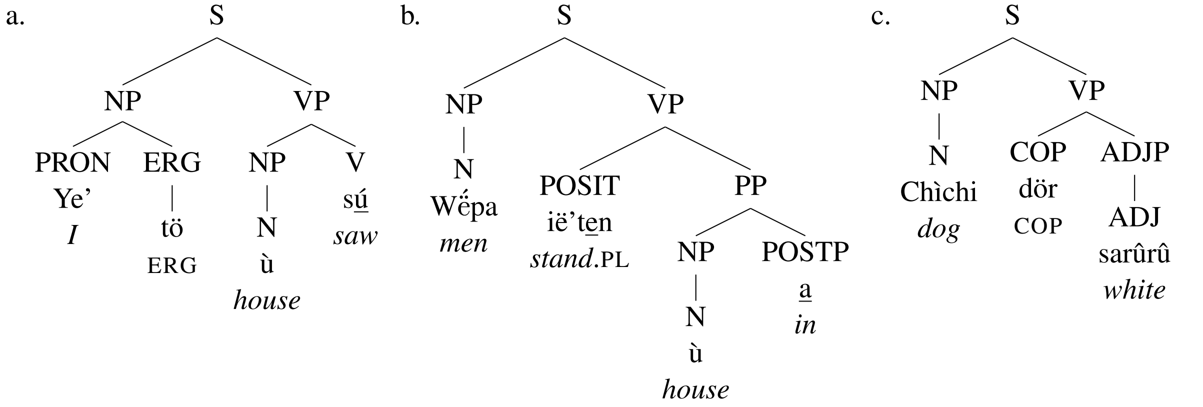
2.3 Constituency Parsing

The next step was the parsing of Bribri. We created an n-ary context-free grammar (CFG) (Chomsky, 1956; Hopcroft and Ullman, 1979) to model Bribri syntax[3], implemented using NLTK in Python (Bird et al., 2009). The grammar contains 122 rules: 10 for sentences, 14 for NPs, 52 for VPs, 23 for terminals and 23 for other non-terminal structures. Example (3) shows example sentences parsed with this grammar[4]. The grammar had to be complemented with filters to reject invalid parsings. For example, the parser rejects sentences where the main verbal phrase doesn't contain a finite verb.

[3] There were early attempts to make transformational grammars of Chibchan languages like Bribri and Cabécar (Bourland, 1976; Wilson, 1986), but most work in Bribri syntax has taken place within the functionalist tradition. There are some works, like Coto-Solano (2009), Coto-Solano et al. (2015) and Pacchiarotti (2016) which have elements of generative theories like Government and Binding and Minimalism.

[4] The current version of the CFG grammar is available at http://github.com/rolandocoto/bribri-cfg.

(3) CFG parses for transitive, intransitive and copular sentences: (a) *Ye' tö ù sú̱* ‘I saw the house’ (Constenla et al., 2004, 52), (b) *Wë́pa ië'te̱n ù a̱* ‘The men are in the house (standing)’ (Constenla et al., 2004, 67) and (c) *Chìchi dör sarûrû* ‘The dog is white’ (Constenla et al., 2004, 60).



This grammar can parse most simple sentences and some complex sentences, such as adverbial clauses and verbal complements. However, there are some complex structures, such as relative clauses, that cannot be parsed by the current iteration of the parser[5]. These sentences were decomposed into simpler structures and then linked together manually into a single CFG tree.

2.4 Dependency Parsing

We used the method of Xia and Palmer (2001) to raise the heads of the CFG subtrees and establish the dependencies between words. We then wrote a series of rules to establish the relations between dependencies; the relations were drawn from version 2.8 of Universal Dependencies, henceforth UD. After this first pass, some parses had to be automatically corrected to match the UD standards. For examples, copular sentences needed to be corrected to make the attribute the head. After setting the relations we converted the Bribri-specific parts of speech to Universal POS tags (UPOS). Several parts of speech were merged into a single UPOS (e.g. verbs and positional verbs were merged into UPOS VERB). Finally, the parser extracted the features of verbs and adverbs with negative polarity. The features of nouns, pronouns and determiners are pending in the current iteration of the parser.

3 Results: Common Structures in Bribri

The methodology described above was used to automatically generate dependency parses for 150 Bribri sentences. Table 1 shows the percentage of UPOS tags in the dataset. The four most common parts of speech, PRON, VERB, NOUN and ADP, account for 73% of the words in the corpus.

PRON	183 (24%)	ADP	68 (9%)	PUNCT	28 (4%)	ADJ	14 (2%)	NUM	5
VERB	163 (21%)	PART	44 (6%)	AUX	26 (4%)	DET	10 (1%)	CCONJ	3
NOUN	146 (19%)	ADV	39 (5%)	PROPN	21 (3%)	SCONJ	7 (1%)	INTJ	3

Table 1: UPOS tags in the Bribri sentences. Counts without percentages are less than 1% of the total.

Table 2 shows the relations found in the corpus. The most common relations, *root* and *nsubj*, account for 40% of the total. There are some relations, like *reparandum*, that are found infrequently, but could become more frequent as the corpus is expanded with conversational data from oral narrations.

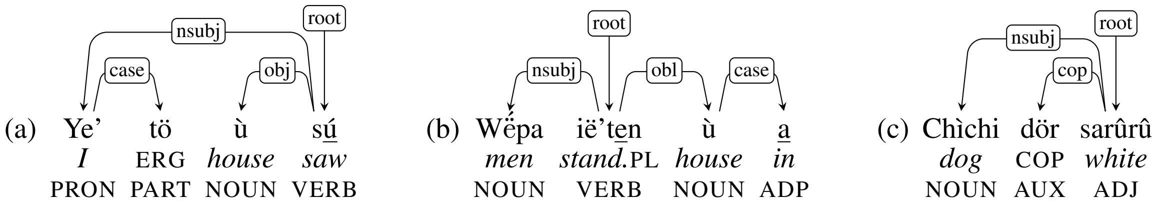
Example (4) shows dependency parses for transitive, intransitive and copular sentences. These are the same sentences that were shown as CFG parses in example (3) above. They show three different objects as roots: a verb (*sú̱* ‘saw’), a positional verb (*ië'te̱n* ‘to be in a place, standing’) and an adjective as the attribute of a copula (*sarûrû* ‘white’). They also show basic relationships such as *nsubj* for ergative and absolutive subjects, *obj* for an absolutive direct object, and *obl* for an oblique argument.

[5] Out of the 150 sentences, 104 (70%) were parsed completely automatically. For 23 of the sentences (15%), the correct POS was provided manually and the CFG and DepParses were generated automatically. For another 23 of the sentences (15%), both the POS tag and the CFG parse were provided manually and the DepParse was generated automatically.

nsubj	154 (20%)	cop	26 (3%)	advcl	8 (1%)	intj	3
root	150 (20%)	nmod	18 (2%)	amod	6	appos	2
case	89 (12%)	nmod:poss	18 (2%)	nummod	5	ccomp	2
obl	78 (10%)	xcomp	16 (2%)	compound	4	fixed	1
advmod	63 (8%)	conj	12 (2%)	acl:recl	3	reparandum	1
obj	46 (6%)	mark	11 (1%)	cc	3		
punct	28 (4%)	det	10 (1%)	flat	3		

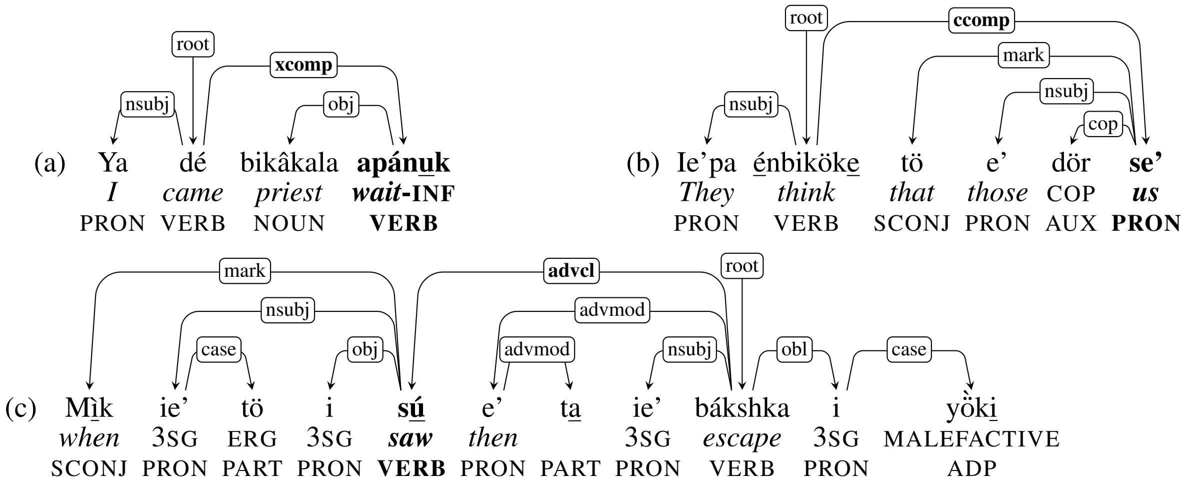
Table 2: Relations in the Bribri sentences. Counts without percentages are less than 1% of the total.

(4) Dependency parse for transitives, intransitives and copulas: (a) *Ye' tö ù sú̱* 'I saw the house' (Constenla et al., 2004, 52), (b) *Wë̃pa ië'te̱n ù a̱* 'The men are in the house (standing)' (Constenla et al., 2004, 67) and (c) *Chìchi dör sarûrû* 'The dog is white' (Constenla et al., 2004, 60).



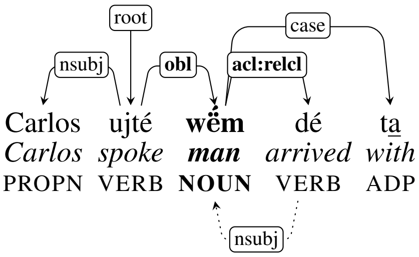
The examples in (5) show more complex sentences. The sentence in (5a) has a clausal complement marked with *xcomp*, the phrase *bikâkala apánu̱k* 'to wait for the master of ceremonies' (a type of priest). The sentence in (5b) has a copular clause as a direct object, and so it is marked with the *ccomp* relation. The sentence in (5c) includes an adverbial clause that precedes the main clause. Therefore, the head of the subclause is connected to the root using the *advcl* relation.

(5) Dependency parse for (a) *Ya dé bikâkala apánu̱k* 'I came to wait for the master of ceremonies (priest)' (Constenla et al., 2004, 47), (b) *Ie'pa é̱nbiköke̱ tö e' dör se'* 'They think that those [spirits] are one of us' (Constenla et al., 2004, 114) and (c) *Mi̱k ie' tö i sú̱ e' ta̱ ie' bákshka i yö̀ki̱* 'When he saw him, he ran away from him' (Constenla et al., 2004, 112).


(a)
root
nsubj
xcomp
obj
Ya dé bikâkala apánu̱k
I came priest wait-INF
PRON VERB NOUN VERB
(b)
root
ccomp
mark
nsubj
nsubj
cop
Ie'pa é̱nbiköke̱ tö e' dör se'
They think that those COP us
PRON VERB SCONJ PRON AUX PRON
(c)
mark
advcl
root
nsubj
advmod
case
obj
advmod
nsubj
obl
case
Mi̱k ie' tö i sú̱ e' ta̱ ie' bákshka i yö̀ki̱
when 3SG ERG 3SG saw then 3SG escape 3SG MALEFACTIVE
SCONJ PRON PART PRON VERB PRON PART PRON VERB PRON ADP

All of the previous examples were parsed automatically by the CFG grammar and then converted automatically into a dependency parse. However, example (6) shows a complex clause that cannot yet be parsed. This is a head-internal relative clause, the main type of relative clause in Bribri (Coto-Solano et al., 2015). The sentence *Carlos ujté* **wë̃m** *dé ta̱* 'Carlos spoke with the man that arrived' has the main verb *ujté* 'spoke' and the relativized verb *dé* 'arrived'. (Bribri does not have an attributive conjugation, so the main and subordinate verbs have the same morphological forms). The head of the relative clause is *wë̃m* 'man', which is an oblique argument to the main verb and the subject of the relativized verb.

(6) Dependency parse for *Carlos ujté wẽ̈m dé ta̲* ‘Carlos spoke with the man that arrived’ (Constenla et al., 2004, 54). It includes an enhanced dependency for the subject of the relative clause.



Because this structure cannot be parsed by the CFG it can’t be converted to UD automatically. We parsed it separately as two clauses, and then joined them manually as a single constituency parse, which was then converted to a dependency parse using the procedure described above. This clause is also noteworthy in that an enhanced dependency was included to mark the relation between the relativized verb and the head of the relative clause. Further research needs to be conduct in order to parse these in a fully automated fashion.

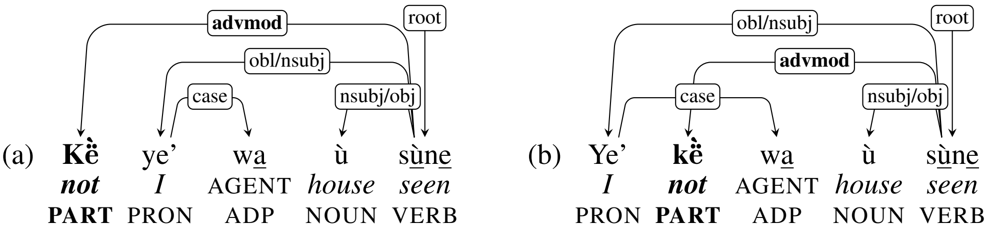
4 Challenging Bribri Structures

There were numerous challenges during the process of dependency parsing. Here we will focus on four of them: (a) structures with flexible order, (b) the treatment of sequences of verbs and positional verbs, (c) the relations of arguments in sentences with middle voice verbs, intransitives of motion and possession, and (d) the differences between UD tense features and the Bribri tense system.

4.1 Flexible word ordering

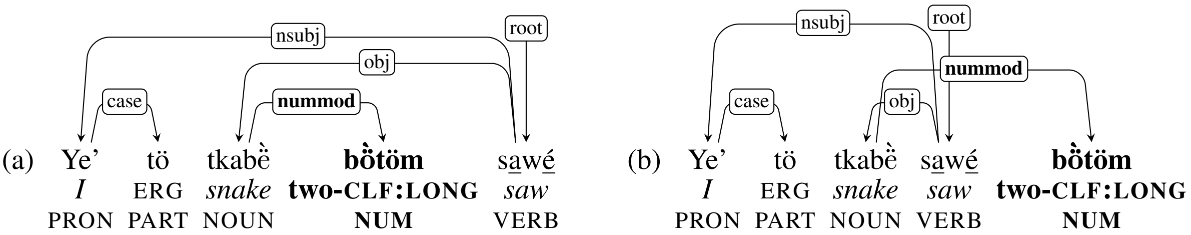
Bribri has several elements that admit flexible word-ordering, which can lead to non-projective parses. One such element is the negative adverb *kë̀* ‘not’. In sentence (7a), the negative is at the edge of the sentence, without interfering with other relations. However, in sentence (7b), the negative particle is between the pronoun *ye’* ‘I’ and its case marker *wa̲*. (For whether the clause with *wa̲* should be labeled as *obl* or *nsubj*, see section 4.3 below).

(7) Dependency parse for (a) *Kë̀ ye’ wa̲ ù sù̲ne̲* ‘I didn’t see the house’ and (b) *Ye’ kë̀ wa̲ ù sù̲ne̲* ‘I didn’t see the house’ (Constenla et al., 2004, 53). The parse in (b) is non-projective.



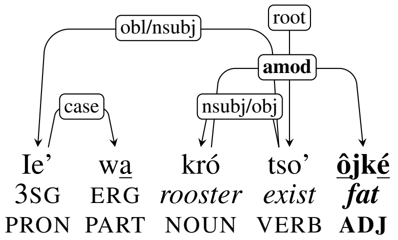
Another flexible structure is found when an absolutive noun is modified by a numeral or an adjective. In example (8a) ‘I saw **the two** snakes’, the noun is immediately followed by the numeral. However, in example (8b), ‘I saw **two** snakes’, the numeral is placed at the end of the sentence, and there is a verb between the noun and its numeral.

(8) Dependency parse for (a) *Ye’ tö tkabë̀ bö̀töm sa̲wé̲* ‘I saw the two snakes’ and (b) *Ye’ tö tkabë̀ sa̲wé̲ bö̀töm* ‘I saw two snakes’ (Constenla et al., 2004, 70). The parse in (b) is non-projective.


nsubj
root
obj
case
nummod
(a) Ye’ tö tkabë̀ bö̀töm sa̲wé̲
I ERG snake two-CLF:LONG saw
PRON PART NOUN NUM VERB
nsubj
root
nummod
case
obj
(b) Ye’ tö tkabë̀ sa̲wé̲ bö̀töm
I ERG snake saw two-CLF:LONG
PRON PART NOUN VERB NUM

Adjectives and participles can also show this behavior. Example (9) shows the adjective *ô̲jké̲* ‘fat’, which describes the noun *kró* ‘rooster’. However, the noun-adjective connection crosses the connections of the root verb with its constituents. The current CFG parser can parse negatives and numerals, but the correct parsing of adjectives separate from their nouns remains for future work.

(9) Non-projective parse for *Ie’ wa̲ kró tso’ ô̲jké̲* ‘She has a fat rooster’ (Pacchiarotti, 2020, 254).


obl/nsubj
root
amod
case
nsubj/obj
Ie’ wa̲ kró tso’ **ô̲jké̲**
3SG ERG *rooster* *exist* ***fat***
PRON PART NOUN VERB **ADJ**

4.2 Positional Verbs as Auxiliaries

Example (10a) shows a sentence with the positional verb *dur* ‘to be in a place, standing’. This positional would be the root of the dependency parse. Example (10b) has a sentence with the negative verb *ku̲* ‘not to be in a place’; this would also be the root of its sentence. However, example (10c) shows a sentence where both of these verbs are in a sequence. Which of the two should be the root?

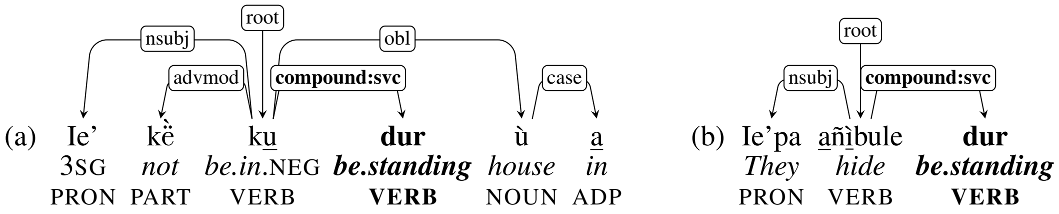
(10) a. Ie’ **dur** ù a̲
3SG **ROOT:*be.standing*** *house in*
‘He is (standing) in the house’ (Constenla et al., 2004, 67)

b. Ie’ kȅ **ku̲** ù a̲
3SG *not* **ROOT:*be.in*.NEG.IPFV** *house in*
‘He is not in the house’ (Constenla et al., 2004, 67)

c. Ie’ kȅ **ku̲** **dur** ù a̲
3SG *not* ***be.in*.NEG.IPFV** ***be.standing*** *house in*
‘He is not (standing) in the house’ (Constenla et al., 2004, 67)

This second verb in this construction is not a light verb because both verbs contribute semantic content to the sentence. It is also not an auxiliary because it contains little or no information about tense, aspect, mood, voice or evidentiality. (These positional verbs do not take the set of TAM suffixes that other verbs do). Therefore, we will treat this sequence as an *asymmetrical serial verb* (Aikhenvald, 2006), where the first verb carries the TAM marking and the second verb contributes motion information to the sentence. We will also follow the analysis of Jara Murillo (2013), Pacchiarotti (2015) and Krohn (2017) and treat the first element of the verb chain as the root of the structure, and the positional verb as the secondary verb. Two examples of these serial structures are shown in (11).

(11) Dependency parse for (a) *Ie’ kȅ ku̲ dur ù a̲* ‘He is not (standing) in the house’ (Constenla et al., 2004, 47) and (b) *Ie’pa a̲ñ̲i̲bule dur* ‘They are hiding (standing)’ (Jara, 2018, 203).

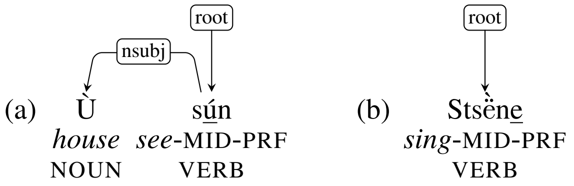

root
nsubj
obl
advmod
compound:svc
case
(a) Ie’ kȅ ku̲ **dur** ù a̲
3SG *not* *be.in*.NEG ***be.standing*** *house* *in*
PRON PART VERB **VERB** NOUN ADP
root
nsubj
compound:svc
(b) Ie’pa a̲ñ̲i̲bule **dur**
They *hide* ***be.standing***
PRON VERB **VERB**

4.3 Core Arguments in Middle Voice and Intransitive Verbs

The marking of the core arguments of verbs is straightforward in most cases. As shown above, the ergative marker can be used to find the *nsubj*, and its presence or absence can be used to determine whether the absolutive is an *nsubj* or *obj*. However, there are structures, like middle voice verbs and some intransitives, where this decision is more complex.

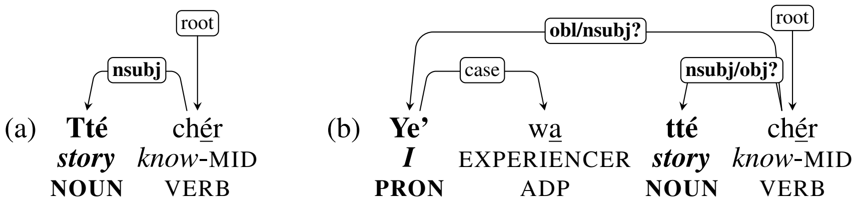
In Bribri middle voice verbs, the subject is usually the patient of the action, and the agent of the action is understood as an unspecified "general" agent. In (12a), *ù sú̲n* ‘houses are visible’, the houses could be "seen" by anyone passing by. In sentence (12b), *stsë̀ne̲* ‘there was singing’, there is no specific person doing the singing. This would be similar to *on chante* or *ça chante* in French, or *man singt* in German.

(12) Dependency parse for (a) *Ù sú̲n* ‘Houses are visible’ (lit: ‘houses are seen’) (Constenla et al., 2004, 84) and (b) *Stsë̀ne̲* ‘There was singing’ (Constenla et al., 2004, 26).



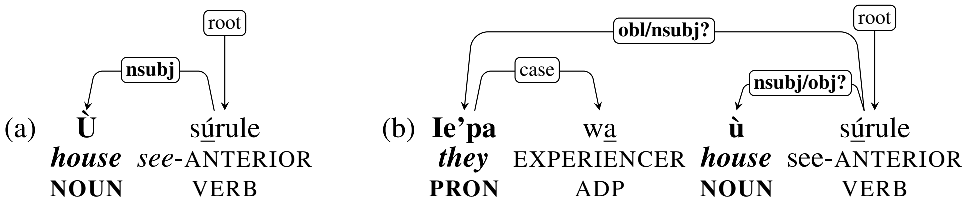
From a morphological point of view, middle verbs are not transitive, and should not be able to take agents. However, middle verbs can add an argument using the postposition *wa̲*. Sentence (13a), ‘The story is known’ is a typical middle voice structure. But sentence (13b) ‘I know the story’ has an additional argument to indicate who is experiencing the knowing of the story. This argument could be described as an oblique, and the noun ‘story’ could be the subject in both sentences. This is a consistent way to describe two verbs with identical morphology. However, there is a second alternative: The phrase *ye’ wa̲* in (13b) could also be described as the ergative of the sentence, which would turn the noun ‘story’ into the direct object (Pacchiarotti and Kulikov, 2021, 4).

(13) Dependency parse for middle voice sentences: (a) *Tté ché̲r* ‘The story is known’ and (b) *Ye’ wa̲ tté ché̲r* ‘I know the story’ (lit: ‘the story is known by me’) (Pacchiarotti, 2016, 6).



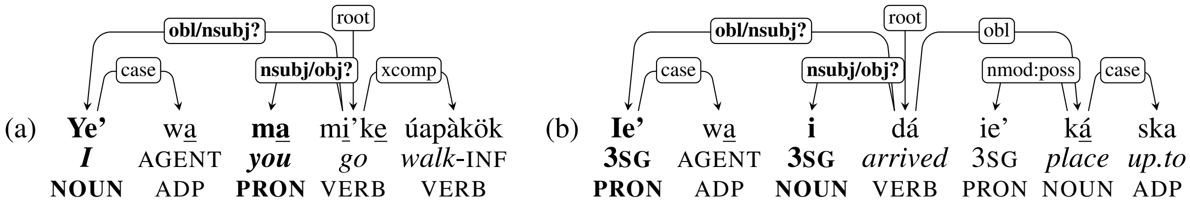
This type of structure, where an argument is added using *wa̲*, is relatively frequent in Bribri. For example, *anterior* verbs (Constenla et al., 2004, 91), also called *antepresent* verbs (Jara, 2018, 72), are a type of pluperfect which are morphologically middle and can be used for middle voice meanings, as in (14a). But anterior verbs can take an additional argument which resembles an ergative, as in (14b).

(14) Dependency parse for anterior verbs, derived from middle voice: (a) *Ù sú̲rule* ‘The house has been seen’ and (b) *Ie’pa wa̲ ù sú̲rule* ‘They have seen the house’ (Constenla et al., 2004, 91).



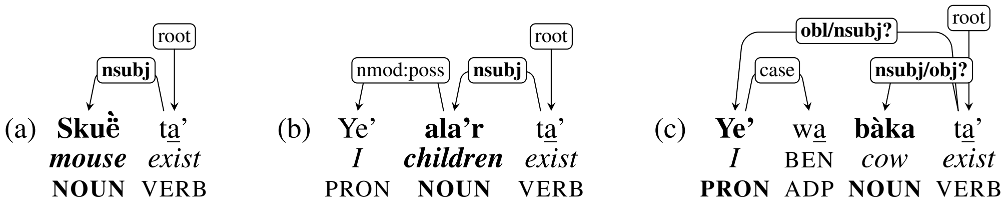
What are the relations between the verb and the arguments in the sentences with *wa̲*? Following a morphological versus a semantic criterion would lead to different decisions. The sentences in (15) show motion verbs which are usually used as intransitives, but that here have an added argument for the person who causes the motion. Here the *wa̲* marks the causer of the movement, and the absolutive indicates the patient that is actually moved. Morphologically these verbs are intransitive, so it would make sense to label the *wa̲*-phrase as an oblique. On the other hand, the arguments are an agent and a patient, so they would resemble a regular ergative phrase, which would call for *nsubj*/*obj* relations coming out of the root.

(15) Dependency parse for sentences of motion: (a) *Ye' wa̲ ma̲ mi̲'ke̲ úapàkök* 'I'll take you for a walk' and (b) *Ie' wa̲ i dá ie' ká̲ ska* 'She took her to her place' (Constenla et al., 2004, 117-118).


obl/nsubj?
root
case
nsubj/obj?
xcomp
(a) Ye' wa̲ ma̲ mi̲'ke̲ úapàkök
I AGENT you go walk-INF
NOUN ADP PRON VERB VERB
obl/nsubj?
root
obl
case
nsubj/obj?
nmod:poss
case
(b) Ie' wa̲ i dá ie' ká̲ ska
3SG AGENT 3SG arrived 3SG place up.to
PRON ADP NOUN VERB PRON NOUN ADP

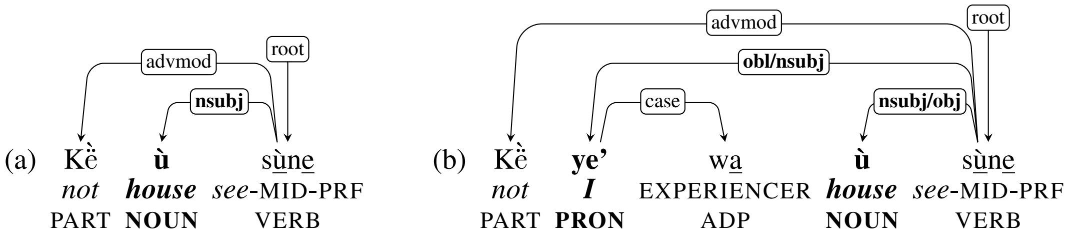
This question about how to tag the arguments of intransitives can also be seen when the verb *ta̲'* 'to exist' is used with alienable possessives. The sentence (16b), *Ye' ala'r ta̲'* 'I have children' has an inalienable possessive as its absolutive subject. Here, the possessor *ye'* 'I' is expressed as a modifier to the absolutive noun *ala'r* 'children'. On the other hand, sentence (16c) has the alienable possessor marked with *wa̲*. The argumentation here would be similar to that of the motion verbs: The verb *ta̲'* 'to exist' is morphologically intransitive and should therefore have only one core argument (the thing possessed), marked with *nsubj*. Moreover, this structure is similar to possessives in languages like Russian, where the possessor is marked with a preposition and the genitive case. On the other hand, the absolutive argument is a theme, so this would again make it a candidate for the *obj* relation.

(16) Dependency parses for existence and possession: (a) *Skuȅ ta̲'* 'There are mice', (b) *Ye' ala'r ta̲'* 'I have children' and (c) *Ye' wa̲ bàka ta̲'* 'I have cows' (Constenla et al., 2004, 53, 74, 105).



The structures that constitute the strongest argument for labelling *wa̲*-phrases as *nsubj* are the transitive negatives. These are constructed using middle verbs, and the agent/experiencer does not receive its usual ergative marker. In sentence (17b), the experiencer is marked with *wa̲* and the theme is marked with the absolutive. In the corresponding positive version of the sentence, *Ye' tö ù sú̲* 'I saw the house' shown in (4a), the experiencer is marked with the ergative *tö* and the theme is again marked with the absolutive. Given the parallels between the two, it could be conceivable to mark the *wa̲*-structure with the *nsubj* relation and the absolutive with *obj* (Margery, 2005; Cruz Volio, 2010; Pacchiarotti, 2016). However, it would be equally useful to consistently mark the absolutive as the subject of the morphologically middle verb, so that both (17a) and (17b) have the word *ù* 'house' as their subject (Constenla et al., 2004; Jara, 1995; Barguigue, 2016).

(17) Dependency parse for (a) *Kȅ ù sù̲ne̲* 'The house isn't seen' and (b) *Ye' kȅ wa̲ ù sù̲ne̲* 'I didn't see the house' (Constenla et al., 2004, 53).



So, which criterion to use, the morphological or the semantic? In the current version of our dependency parser we have chosen the relations to be consistent with the morphology of the verbs, and so the absolutives of intransitive and middle voice verbs are marked as *nsubj*, and the other arguments are marked as *obl*. Further investigation into other syntactic properties of subjects is needed, and therefore the exact relations of these verbs could change in future iterations of the parser. One potential solution would be to mark the arguments as *obl*/*nsubj* in the dependencies and to use enhanced dependencies to further mark them as semantic *nsubj*/*obj* (Przepiórkowski and Patejuk, 2020).

4.4 Bribri Tenses and Universal Features

The Universal Feature system in UD includes the values {Past, Pres, Fut}[6] for the Tense feature. However, Bribri morphology does not match these categories, which makes the automatic extraction of features complex. The main verbal distinction in Bribri is aspect. It has perfect and imperfect verbs, and this does match the feature system. However, tense splits verbs in different ways. The temporal point of split between tenses is “the sunset of the night before" (Constenla et al., 2004, 15). This splits time into two tenses: the *remote* tense and the *recent* tense. The remote tense refers to actions that take place before yesterday’s sunset, while the recent tense includes actions done in the recent past (e.g. today’s morning), in the present (right now) and in the near future (e.g. "soon"). Table 3 shows examples of how these tenses interact with the aspect system. The *remote* tense is not problematic for automatic parsing, given that their UD tense will always be Past and their aspect can be determined from their morphology.

<table>
<tr><td>Aspect</td><td>UF Tense</td><td>Past</td><td>Past</td><td>Present</td><td>Future</td><td>Future</td></tr>
<tr><td></td><td>Bribri tense</td><td>Remote</td><td>(today)</td><td>Present</td><td>(near future)</td><td>Future</td></tr>
<tr><td>Perfect</td><td></td><td>Perfect remote
ya’
‘drank’</td><td colspan="2">Perfect recent
yé
‘drinks’, ‘drank’</td><td colspan="2"></td></tr>
<tr><td rowspan="3">Imperfect</td><td rowspan="3"></td><td rowspan="3"></td><td colspan="2">Imperfect recent
yè
‘drinks’, ‘was drinking’</td><td></td><td>Certain future
yèrâ
‘will drink’</td></tr>
<tr><td colspan="3">Durative
yèke
‘drinks’, ‘used to drink’, ‘shall drink’</td><td></td></tr>
<tr><td></td><td colspan="3">Future potential
yèmi̱
‘can drink’, ‘shall drink’</td></tr>
</table>

Table 3: Examples of interaction between Bribri and the current version of Universal Features (UF) tenses in active voice verbs

The main issue comes with the verbs in the *recent* tense. This Bribri tense is similar to the *hodiernal* tense in Mwotlap (François, 2003), Haya, Luganda and Ancash Quechua (Comrie, 1985), in that the recent tense includes actions that have happened “today”, regardless of whether they are in the past or in the near future. Depending on the context, the verbs in the recent tense could overlap with several of the time categories in Universal Features. For example, the imperfect recent form *yè* includes events that have happened before the present moment and simultaneous with the present moment, so this could be translated as ‘drinks’ or ‘was drinking’. A sentence like *Ye’ yè* could be translated as ‘I drink it’ or ‘I was drinking it’. Without any contextual cues, it wouldn’t be possible to automatically determine the appropriate tense in the Universal Feature system. There are other verbal forms, such as the *future potential* (Jara, 2018, 73), also called the *imperfect potential* (Constenla et al., 2004, 111), that also spread across two tenses of Universal Features. The sentence *Yì ki̱ be’ kiàrmi̱?* (Jara, 2018, 73) can be translated as either a potential in the present tense, ‘Who can love you?’, or an imperfect future tense, ‘Who shall love you?’. In this sentence there are no cues to aid the automatic parsing in selecting between the Tense=Pres and the Tense=Fut features.

[6] There are more Bribri verb forms that those mentioned here, and they include verbs in other tense categories of Universal Features. For example, the perfect antepresent form *yéule* ‘to have drunk’ would be marked with the feature Tense=Pqp.

The problem is even more pronounced with verbs in the *durative/habitual* form (Jara, 2018, 74), also called the *habitual imperfect* (Flores-Solórzano, 2017b, 34) and the *second imperfective* (Constenla et al., 2004, 90). The sentence *Ye' kanè̠blö̀ke* (Jara, 2018, 74) has an imperfect aspect, but it is spread across the recent tense. It can be translated as ‘I used to work’ in the recent past, ‘I regularly work’ in the habitual/present and ‘I shall work soon’ in the near future. In this sentence the tense feature could take three different values (Past, Pres, Fut), without a way to automatically distinguish between them using only the words in the sentence. One potential solution would be to leave the tense feature out of the description of these verbs, and add an annotation of their tense in the MISC field of the CONLL-U file. Another solution would be to add a feature such as Tense=Hod to the Universal Feature system, which would allow for a richer and more cross-linguistically faithful analysis of the UD database as a whole.

5 Conclusions and Future Work

This paper presents a first attempt to parse Bribri sentences using context-free grammars and dependency grammars, and it presents an adaptation of Universal Dependencies to Bribri. This preliminary effort illustrates the possibility of applying UD to Chibchan languages, but also the numerous challenges involved in implementing the task of automated parsing in Indigenous languages. In many ways these languages test the "U" in UD, and we hope that, by embracing languages where there aren't yet optimal solutions or linguistic consensus about their structures, this will help push the endeavor of Universal Dependencies forward. In future work we will expand to corpus to create a first treebank for Bribri and improve the parsers with the ultimate goal of releasing them for public use. We also seek to gather enough Bribri data in CONLL-U format so that we can train deep-learning based parsing methods like *UDPipe 2* (Straka, 2018), which might further accelerate the development of the treebank.

The parsing process presented here is done in the hope of developing tools that might be useful for the documentation and revitalization of Bribri. These should include NLP tools like chatbots and question answering systems, as well as linguistic tools like learning materials, exercises for students of Bribri, and more detailed documentation of the grammar of the language. One major challenge is to expand the process of annotation to include native speakers of Bribri. This would entail expanding the annotation process to non-automated tools, such as the manual annotation interfaces *UD Annotatrix* (Tyers et al., 2018) and *TrED* (Pajas and Fabian, 2000). Finally, we acknowledge the issues of data sovereignty with this work (i.e. non-Bribri researchers working on Bribri data). We have limited ourselves to data that is already publicly available, and in the future, we hope to expand the conversation with Bribri partners to ensure that the creation of NLP tools provides tangible benefits to Bribri partners and to the Bribri community in general.

References

Alexandra Y Aikhenvald. 2006. Serial Verb Constructions in Typological Perspective. *Serial Verb Constructions: A Cross-Linguistic Typology*, pages 1–68.

Saïd Barguigue. 2016. Predicados Afectivos en el Bribri: Un Acercamiento Tipológico-Funcional. Master's thesis, Universidad de Sonora.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

David Hawley Bourland. 1976. Una gramática generativa-transformacional del cabécar. *Revista de Filología y Lingüística de la Universidad de Costa Rica Vol. 2 Núm. 3*, pages 49–100.

Emily Chen, Hyunji Hayley Park, and Lane Schwartz. 2020. Improved Finite-State Morphological Analysis for St. Lawrence Island Yupik using Paradigm Function Morphology. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2676–2684.

Noam Chomsky. 1956. Three Models for the Description of Language. *IRE Transactions on information theory*, 2(3):113–124.

Bernard Comrie. 1985. *Tense*, volume 17. Cambridge University Press.

Adolfo Constenla, Feliciano Elizondo, and Francisco Pereira. 2004. *Curso Básico de Bribri*. Editorial de la Universidad de Costa Rica.

Rolando Coto-Solano and Sofía Flores-Solórzano. 2016. Alineación forzada sin entrenamiento para la anotación automática de corpus orales de las lenguas indígenas de costa rica. *Kánina*, 40(4):175–199.

Rolando Coto-Solano and Sofía Flores-Solórzano. 2017. Comparison of Two Forced Alignment Systems for Aligning Bribri Speech. *CLEI Electron. J.*, 20(1):2–1.

Rolando Coto-Solano, Adriana Molina-Muñoz, and Alí García Segura. 2015. Correlative Structures in Bribri. *University of British Columbia Working Papers in Linguistics*, 43:27–41.

Rolando Coto-Solano. 2009. Reanálisis de las cláusulas relativas en la lengua bribri como un caso de linearización en la teoría minimalista. Memoria del II Congreso Internacional de Lingüística Aplicada (CILAP).

Rolando Coto-Solano. 2015. The Phonetics, Phonology and Phonotactics of the Bribri Language. In *2nd International Conference on Mesoamerican Linguistics. California State University, Los Angeles*.

Rolando Coto-Solano. 2021. Explicit Tone Transcription Improves ASR Performance in Extremely Low-Resource Languages: A case study in Bribri. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 173–184, Online, June. Association for Computational Linguistics.

Gabriela Cruz Volio. 2010. El sistema de transitividad en las cláusulas materiales del bribri según la gramática sistémico-funcional. *Revista de Filología y Lingüística de la Universidad de Costa Rica*, pages 133–154.

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

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir, et al. 2021. AmericasNLI: Evaluating Zero-shot Natural Language Understanding of Pretrained Multilingual Models in Truly Low-Resource Languages. *arXiv preprint arXiv:2104.08726*.

Isaac Feldman and Rolando Coto-Solano. 2020. Neural Machine Translation Models with Back-Translation for the Extremely Low-Resource Indigenous Language Bribri. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3965–3976.

Fabrício Ferraz Gerardi, Stanislav Reichert, Carolina Aragon, Lorena Martín-Rodríguez, Gustavo Godoy, and Tatiana Merzhevich. 2021. Tudet: Tupían Dependency Treebank (version v0.2).

Sofía Flores-Solórzano. 2010. Teclado Chibcha: Un software lingüístico para los sistemas de escritura de las lenguas bribri y cabécar. *Revista de Filología y Lingüística de la Universidad de Costa Rica*, pages 155–161.

Sofía Flores-Solórzano. 2017a. Corpus oral pandialectal de la lengua bribri. `http://bribri.net`.

Sofía Margarita Flores-Solórzano. 2017b. *Un primer corpus pandialectal oral de la lengua bribri y su anotación morfológica con base en el modelo de estados finitos*. Ph.D. thesis, Universidad Autónoma de Madrid.

Sofía Flores-Solórzano. 2019. La modelización de la morfología verbal bribri - Modeling the Verbal Morphology of Bribri. *Revista de Procesamiento del Lenguaje Natural*, 62:85–92.

Alexandre François. 2003. *La sémantique du prédicat en Mwotlap, Vanuatu*, volume 84. Peeters Publishers.

Alí García Segura. 2016. *Ditsö Rukuö Identity of the Seeds: Learning from Nature*. IUCN.

Andrew Garrett, Clare Sandy, Erik Maier, Line Mikkelsen, and Patrick Davidson. 2013. Developing the Karuk Treebank. In *Fieldwork Forum, Department of Linguistics, UC Berkeley*.

John E Hopcroft and Jeffrey D Ullman. 1979. Introduction to Automata Theory, Languages and Computation. *Adison-Wesley*.

Mans Hulden. 2009. Foma: A Finite-State Compiler and Library. In *Proceedings of the Demonstrations Session at EACL 2009*, pages 29–32.

INEC. 2011. Población total en territorios indígenas por autoidentificación a la etnia indígena y habla de alguna lengua indígena, según pueblo y territorio indígena. In Instituto Nacional de Estadística y Censos, editor, *Censo 2011*.

Carla Victoria Jara Murillo. 2013. Morfología verbal de la lengua bribri. *Estudios de Lingüística Chibcha*.

Carla Victoria Jara. 1995. Transitividad en el discurso bribri. *Revista de filología y lingüística de la Universidad de Costa Rica*, 21(2):93–105.

Carla Victoria Jara. 2018. *Gramática de la lengua bribri*. E-Digital ED.

Haakon S Krohn. 2017. Semántica de los posicionales del bribri. *Revista de Filología y Lingüística de la Universidad de Costa Rica*, 43(1):117–136.

Haakon Krohn. 2020. Elaboración de una base de datos en XML para un diccionario bribri–español español–bribri en la web. *Porto das Letras*, 6(3):38–58.

Haakon S. Krohn. 2021. Diccionario digital bilingüe bribri. `http://www.haakonkrohn.com/bribri`.

Manuel Mager, Arturo Oncevay, Abteen Ebrahimi, John Ortega, Annette Rios Gonzales, Angela Fan, Ximena Gutierrez-Vasques, Luis Chiruzzo, Gustavo Giménez Lugo, Ricardo Ramos, et al. 2021. Findings of the AmericasNLP 2021 Shared Task on Open Machine Translation for Indigenous Languages of the Americas. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 202–217.

Enrique Margery. 2005. *Diccionario fraseológico bribri-español español-bribri*. Editorial de la Universidad de Costa Rica, second edition.

William B McGregor. 2009. Typology of Ergativity. *Language and Linguistics Compass*, 3(1):480–508.

MEP. 2017. *Los Bribri y Cabécares de Sulá, Tomo 1 - Minienciclopedia de los Territorios Indígenas de Costa Rica*. Dirección de Desarrollo Curricular, Educación Intercultural. Ministerio de Educación Pública.

Christopher Moseley. 2010. *Atlas of the World's Languages in Danger*. Unesco.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Sara Pacchiarotti and Leonid Kulikov. 2021. Bribri Media Tantum Verbs and the Rise of Labile Syntax. *Linguistics*.

Sara Pacchiarotti. 2015. The Argument Structure of some Caused Motion Constructions in Bribri: A Possible Explanation. In *18th Workshop on American Indigenous Languages (WAILS)*.

Sara Pacchiarotti. 2016. Verbal Deponency in the Chibchan Family. In *49th Annual Meeting of the Societas Linguistica Europaea*.

Sara Pacchiarotti. 2020. On the Origins of the Ergative Marker wã in the Viceitic Languages of the Chibchan Family. In *Reconstructing Syntax*, pages 241–288. Brill.

Petr Pajas and P Fabian. 2000. Tree Editor TrED, Prague Dependency Treebank, Charles University, Prague. *See URL http://ufal. mff. cuni. cz/˜ pajas/tred*.

Hyunji 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.

Adam Przepiórkowski and Agnieszka Patejuk. 2020. From Lexical Functional Grammar to Enhanced Universal Dependencies. *Language Resources and Evaluation*, 54(1):185–221.

J Diego Quesada. 1999. Ergativity in Chibchan. *STUF-Language Typology and Universals*, 52(1):22–51.

Annette Rios, Anne Göhring, and Martin Volk. 2008. A Quechua-Spanish Parallel Treebank. *LOT Occasional Series*, 12:53–64.

Jack Rueter, Marília Fernanda Pereira de Freitas, Sidney da Silva Facundes, Mika Hämäläinen, and Niko Partanen. 2021. Apurinã Universal Dependencies Treebank. *arXiv preprint arXiv:2106.03391*.

Carlos Sánchez Avendaño. 2013. Lenguas en peligro en Costa Rica: vitalidad, documentación y descripción. *Revista Káñina*, 37(1):219–250.

Justin Spence, Zoey Liu, Kayla Palakurthy, and Tyler Lee-Wynant. 2018. Syntactic Annotation of a Hupa Text Corpus. Technical report, Working Papers in Athabaskan Languages: Alaska Native Language Center ....

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium, October. Association for Computational Linguistics.

Guillaume Thomas. 2019. Universal Dependencies for Mbyá Guaraní. In *Proceedings of the third workshop on universal dependencies (udw, syntaxfest 2019)*, pages 70–77.

Francis Tyers and Robert Henderson. 2021. A Corpus of K'iche' Annotated for Morphosyntactic Structure. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 10–20.

Francis M. Tyers, Mariya Sheyanova, and Jonathan North Washington. 2018. Ud annotatrix: An annotation tool for universal dependencies. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories (TLT16)*, pages 10–17.

Universal Dependencies Contributors. 2021. reparandum: overridden disfluency.

Petronila Valengana Valengana and Sofía Flores-Solórzano. 2017. Wès sa' tsiru' chkà alèke - Cómo se prepara el cacao dulce. `https://bribri.net/B09h22m53s05sep2012.html`.

Alonso Vasquez, Renzo Ego Aguirre, Candy Angulo, John Miller, Claudia Villanueva, Željko Agić, Roberto Zariquiey, and Arturo Oncevay. 2018. Toward Universal Dependencies for Shipibo-Konibo. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 151–161.

Irina Wagner, Andrew Cowell, and Jena D Hwang. 2016. Applying Universal Dependency to the Arapaho Language. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 171–179.

Jack L Wilson. 1986. Sobre la definición lingüística: el sujeto y el ergativo. *Estudios de Lingüística Chibcha*, pages 59–84.

Fei Xia and Martha Palmer. 2001. Converting Dependency Structures to Phrase Structures. Technical report, Pennsylvania Univ. Philadelphia.

Bootstrapping Role and Reference Grammar Treebanks via Universal Dependencies

Kilian Evang and **Tatiana Bladier** and **Laura Kallmeyer** and **Simon Petitjean**
Heinrich Heine University Düsseldorf
Universitätsstraße 1, 40225 Düsseldorf, Germany
{evang,bladier,kallmeyer,petitjean}@phil.hhu.de

Abstract

We describe ud2rrg, a rule-based approach for converting UD trees to Role and Reference Grammar (RRG) structures. Our conversion method aims at facilitating the annotation of multilingual RRG treebanks. ud2rrg uses general and language-specific conversion rules. In order to evaluate ud2rrg, we approximate the subsequent annotation effort via measures of tree edit distance. Our evaluation, based on English, German, French, Russian, and Farsi, shows that the ud2rrg transformation of UD-parsed data constitutes a highly useful starting point for multilingual RRG treebanking. Once a sufficient amount of data has been annotated in this way, the automatic conversion can be replaced by a statistical parser trained on that data for an even better starting point.

1 Introduction

Role and Reference Grammar (RRG) (Van Valin Jr. and LaPolla, 1997; Van Valin Jr., 2005) is a grammar theory for natural language that shares with Universal Dependencies (Nivre et al., 2016; Nivre et al., 2020) the aim of being descriptively adequate across typologically diverse languages while reflecting their commonalities in its analyses. It also shares with UD a number of design characteristics, such as recognizing dissociated nuclei and the principle to "annotate what is there", eschewing the use of empty elements, cf. de Marneffe et al. (2021). In addition, RRG's separation between constituent structure and operator structure (the latter reflecting the attachment of functional elements) offers an explanatory framework for certain word-order and semantic phenomena. In recent years, the computational linguistics community has become increasingly interested in RRG and has started to formalize RRG (Osswald and Kallmeyer, 2018) and to build resources and tools to support data-driven linguistic research within RRG (Bladier et al., 2018; Bladier et al., 2020; Chiarcos and Fäth, 2019).

As illustrated in the examples in Figure 1, an important feature of RRG is the layered structure of the clause. The nucleus (NUC) contains the (verbal) predicate, arguments attach at the core (CORE) layer, and extracted arguments at the clause (CLAUSE) layer. Each layer also has a periphery, where adjuncts attach (marked PERI). Operators (closed-class elements encoding tense, modality, aspect, negation, etc.) attach at the layer over which they take scope. They are assumed to be part of a separate projection, but we collapse both projections into a single tree structure for convenience. Elements like *wh*-words in English are placed in the pre-core slot (PrCS), and the pre-clausal elements like fronted prepositional or adverbial phrases are placed in the pre-detached position (PrDP).

In this paper we describe the ongoing effort to build RRGparbank[1], a novel-length parallel RRG treebank for English, German, French, Russian, and Farsi, in a semi-automatic fashion. We focus on the automatic part. Exploiting an off-the-shelf UD parser, the text (George Orwell's novel *1984* and translations) is parsed into UD. Then, exploiting structural similarities between UD and RRG, the UD trees are automatically converted into RRG trees (§2). This conversion accelerates the process of manually annotating the corpus (§3). Once enough data has been collected in this way, we replace the rule-based conversion with a statistical RRG parser trained on the collected data. A series of experiments shows that the statistical RRG parser offers a better starting point for annotating once approximately 2 000 sentences

[1] https://rrgparbank.phil.hhu.de

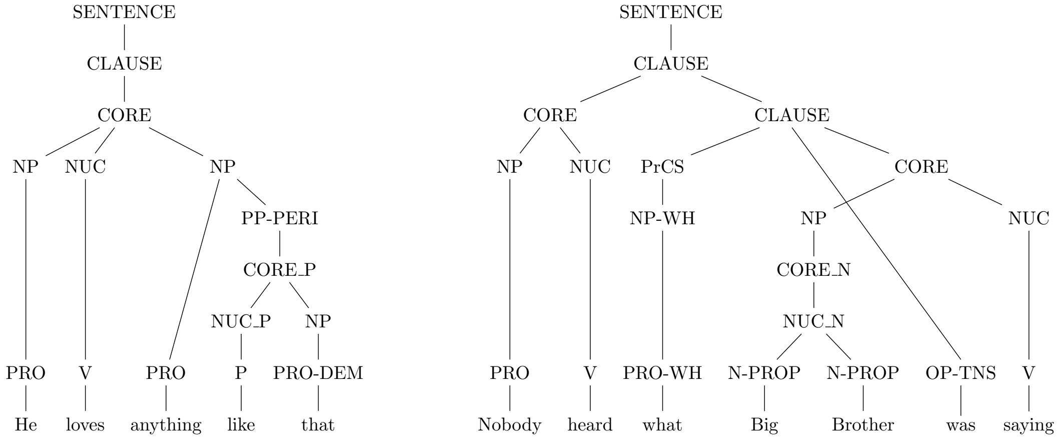


Figure 1: Examples of RRG annotation. Punctuation marks are omitted.

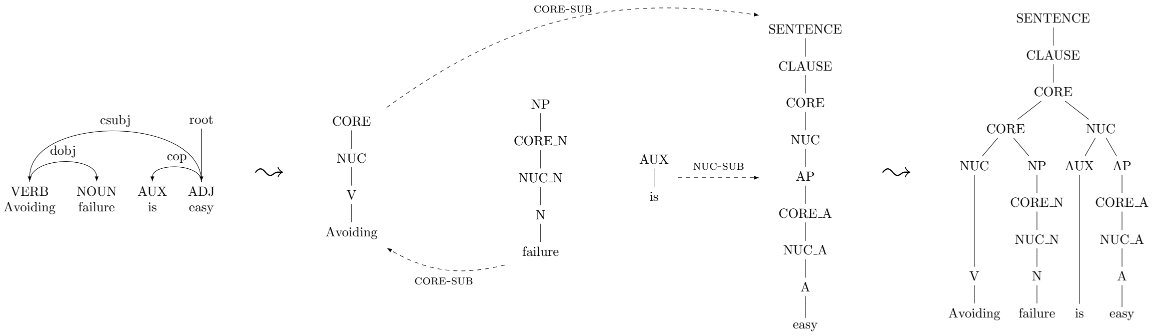


Figure 2: UD tree, RRG derivation tree and resulting RRG tree for the sentence *Avoiding failure is easy*.

are available for training ($4). Finally, we give a qualitative comparison between our converter and that of Chiarcos and Fäth (2019), which targets a slightly different flavor of RRG (§5).

2 UD to RRG Conversion

2.1 Auxiliary Formalism

We define a custom formalism, inspired by tree grammar formalisms such as LTAG (Joshi and Schabes, 1997) that allows us to treat RRG trees as being composed from lexically anchored *elementary trees* via a number of composition operations. Figure 2 and Figure 3 show examples: in the middle, there are a number of elementary trees and the operations with which they are combined (the derivation) and on the right there is the resulting RRG tree. The set of operations is RRG-specific:

- NUC-SUB: presupposes that the host tree is clausal.[2] If the guest tree is clausal, attach its NUC node under the host tree's NUC node, and merge its CORE, CLAUSE and SENTENCE nodes (if any) into the corresponding nodes of the host tree.[3] If not, attach its root under the host tree's NUC node.
- NUC-COSUB: presupposes that both trees are clausal. Create a NUC node above the host tree's NUC node (if it doesn't exist yet), and attach the guest tree's NUC node to that. Merge the CORE, CLAUSE, and SENTENCE nodes.
- NUC-COORD: presupposes that both trees are clausal. Attach the guest tree's NUC node under the host tree's CORE node. Merge the CORE, CLAUSE, and SENTENCE nodes.

[2] A *clausal* tree is an elementary tree whose lexical anchor is the head of a clause. Its spine starts with SENTENCE, CLAUSE, CORE, or NUC, followed by nodes for the lower clausal layers.

[3] By merging we mean attaching any children of the former under the latter.

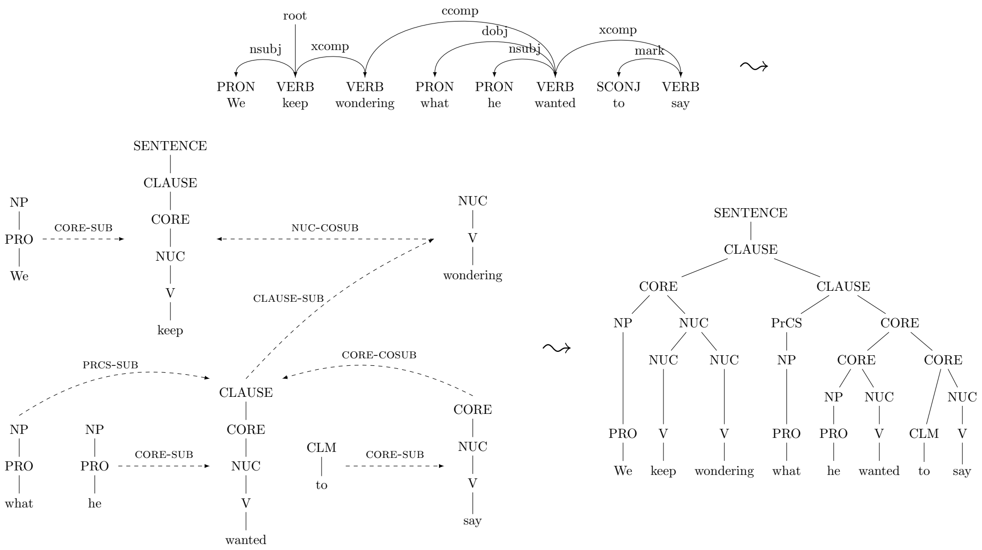


Figure 3: UD tree, RRG derivation tree and resulting tree for the sentence *We keep wondering what he wanted to say*.

- CORE-SUB: presupposes that the host tree is clausal. If the guest tree is clausal, attach its CORE node under the host tree's CORE node, and merge the CLAUSE and SENTENCE nodes (if any). If not, attach its root under the host tree's CORE node.
- CORE-COSUB: presupposes that both trees are clausal. Create a CORE node above the host tree's CORE node (if it doesn't exist yet) and attach the guest tree's CORE node to that. Merge the CLAUSE and SENTENCE nodes.
- CORE-COORD: presupposes that both trees are clausal. Attach the guest tree's CORE node under the host tree's CLAUSE node. Merge the CLAUSE and SENTENCE nodes.
- CLAUSE-SUB: presupposes that the host tree is clausal. If the guest tree is clausal, attach its CLAUSE node under the host tree's CLAUSE node, and merge the SENTENCE nodes (if any). If not, attach its root under the host tree's CLAUSE node.
- PRCS-SUB: create a left PrCS daughter of the host tree's CLAUSE node (if it doesn't exist yet) and attach the guest tree's root under it.
- PRDP-SUB: create a left PrDP daughter of the host tree's SENTENCE node (if it doesn't exist yet) and attach the guest tree's root under it.
- WRAP: attach the host under a designated node (marked $\star$) of the guest (used for attaching preposition complements under PPs).

2.2 General Conversion Rules

If we view the derivations, as exemplified in Figures 2 and 3, as *derivation trees* where nodes are labeled with elementary trees and edges are labeled with operations, then this derivation tree is isomorphic to a corresponding UD tree. What remains to do to convert UD trees to RRG trees is to specify a set of rules that relabel nodes in UD trees with RRG elementary trees, and edges with operations. We try to keep these rules as local as possible, ideally looking only at one UD node and its incoming edge at a time, so a simple recursive traversal of the UD tree suffices. However, as we will see, in some cases we need to take a little more context into account. Table 1 shows the rules used in the example conversions.

UD's content-word-centric approach is a good fit for the conversion to RRG regarding, e.g., copulas, modal, tense, and aspect operators, which RRG treats not as heads of verb phrases but as additional

	label	POS	additional conditions	elementary tree	operation
1	root	ADJ		(SENTENCE (CLAUSE (CORE (NUC (AP (CORE_A (NUC_A (A <>))))))))	
2	cop	AUX		(AUX <>)	NUC-SUB
3	csubj	VERB		(CORE (NUC (V <>)))	CORE-SUB
4	dobj	NOUN		(NP (CORE_N (NUC_N (N <>))))	CORE-SUB
5	root	VERB		(SENTENCE (CLAUSE (CORE (NUC (V <>)))))	
6	nsubj	PRON		(NP (PRO <>))	CORE-SUB
7	dobj	WP or WP$		(NP (PRO <>))	PRCS-SUB
8	mark	SCONJ		(CLM <>)	CORE-SUB
9	ccomp	VERB		(CORE (NUC (V <>)))	CORE-SUB
10	ccomp	VERB	verb of cognition/saying	(CLAUSE (CORE (NUC (V <>))))	CLAUSE-SUB
11	xcomp	VERB		(CORE (NUC (V <>)))	CORE-COSUB
12	xcomp	VERB	phase verb	(NUC (V <>))	NUC-COSUB
13	xcomp	VERB	raising verb	(CORE (NUC (V <>)))	CORE-COORD
14	case	ADP		(PP (CORE_P⋆ (NUC_P (P <>))))	WRAP

Table 1: Examples of rules. Rules 1–12 are the ones used in Figures 2 and 3. Rules 7, 10, 12, and 13 are examples of rules whose implementation requires language-specific POS tags or lexicons.

operators attaching to clauses, cores, or nuclei. By contrast, prepositions are treated as heads of PPs and thus necessitate a slightly more complicated rule (WRAP) to wrap the prepositional complement in a PP. Overall, RRG's approach can be characterized as more content-word-centric than function-word-centric. This and the ready availability of UD resources made UD a more natural starting point for our conversion project than more function-word-centric variants such as SUD (Gerdes et al., 2018).

2.3 Special Conversion Rules

Rules can also make reference to lexical and other language-specific knowledge. One area where this is important is clause linkage, which in UD is always represented with `conj`, `ccomp`, or `xcomp` relations, but in RRG splits up into a more fine-grained set of *juncture-nexus* types. For example, while we subordinate clausal complements at the CORE level by default (Rule 9 in Table 1), clausal complements of verbs of cognition and saying typically require subordination at the CLAUSE level (Rule 10). Similarly, while we cosubordinate open clausal complements at the CORE level by default (Rule 11), open clausal complements of phase verbs as in *starts walking*, *keep wondering* or *stopped believing* require cosubordination at the NUC level (Rule 12). This is illustrated in Figure 3. We have so far implemented this rule for English and for German. For English, we determine the verb class by lookup in the VerbNet lexical database (Kipper-Schuler, 2005). For German, as far as we are aware, similar resources such as GermaNet (Hamp and Feldweg, 1997), provide sets of verb classes which are less fine-grained. Therefore, the equivalent conversion rule for German uses a handwritten set of verbs instead of a lexical database. We have also defined rules to recognize `xcomp` instances encoding the raising construction and trigger core coordination by its associated lemma *seem* in English or *scheinen* in German (Rule 13). Due to the low frequency of relevant phenomena and the inevitable brittleness of rules, we have left these specialized conversion rules as a proof-of-concept and not aimed for more extensive coverage, relying on statistical predictions (see below) and annotators instead for the purpose of building RRGparbank.

2.4 Implementation and Workflow

The text basis for RRGparbank is provided by George Orwell's novel *1984* as well as translations to German, French, Russian and Farsi. The English and Farsi texts, their segmentation into sentences and tokens as well as POS tags and lemmas are taken from the MULTEXT-East dataset (Erjavec, 2017), which also provides the (non-annotated) Russian text. The French and German data was built using the Orwell (1972) and Orwell (2003) editions, respectively. A large part of the German data was annotated by hand following the guidelines of the MULTEXT-East dataset. We used UDpipe2 (Straka, 2018) for segmentation, tagging, and lemmatization of the Russian, French and the non-annotated German data. UD parses for all languages are also provided by UDPipe2.[4]

[4] The reported Labeled Attachment Scores for the 5 languages on UD treebanks are as follows: 85.8% for English, 81.2% for German, 84.3% for Farsi, 83.5% for French and 85.3% for Russian.

Timestamp	nBURP	LF1	failed
#1	0.66	61.02	1 100
#2	0.57	64.09	773
#3	0.47	68.75	355
#4	0.33	72.51	221
#5	0.20	79.96	0

Table 2: nBURP and LF1 scores for the output of ud2rrg using Russian data (4 635 sentences), at different steps of development. Sentences that could not be converted are replaced with flat dummy trees.

In the next step, we use a script called ud2rrg[5], which we developed based on the formalism described above, to convert the UD trees to RRG. It performs a traversal of each UD tree and at each node applies the matching rule, thereby gradually building up an RRG tree. In the rare cases where conversion fails for a node because there is no matching rule (e.g., with rare combinations of POS and grammatical relation), conversion fails and a dummy tree is generated where all tokens are attached to the root.

For the results reported in this paper, 13 annotators with training in RRG annotated 5 453 English, 5 723 German, 2 177 French, 4 675 Russian, and 1 110 Farsi sentences over a time period of 21 months.[6] They were provided the output of ud2rrg and corrected the trees using a graphical interface. The graphical interface and the annotation guidelines were based on RRGbank (Bladier et al., 2018). Development of ud2rrg was ongoing during this period and informed by manual inspection of sentences that failed to convert and of changes annotators made to the ud2rrg output.

Annotation on different languages started at different times. We used the same ud2rrg for all languages, but each new language typically brings with it a number of POS tags and constructions that have not or not much been seen in the data so far, meaning that rules have to be refined and added before ud2rrg performs as well on the new as on the old languages. As an example, consider the case of Russian. Table 2 shows the performance of ud2rrg on Russian at different points in time after its introduction as a new language. We measure the performance in nBURP (smaller is better), LF1 (larger is better), and number of failed sentences (smaller is better) – details are given below in Section 3. Timestamp #1 corresponds to the introduction of the Russian data in the annotation interface. Between #1 and #2, ongoing development of ud2rrg took into account, among other data, Russian gold data produced by annotators. The first rules making specific reference to Russian lexemes were added between #2 and #3 (7 new rules and 2 extensions of existing rules), leading to significantly better performance. Timestamp #4 is a week after #3, while #5 is the time of redaction of this article. The scores keep improving with time, as regular evaluations on the updated gold data indicates which transformation rules are missing, or need an update. Annotators are also encouraged to report sentences for which the transformation is problematic, or fails.

As of this writing, ud2rrg contains about 278 rules, 4 of which depend on language-specific semantic lexical resources to select a juncture-nexus type. In addition, we have language-specific routines to determine finer-grained parts of speech for function words, such as negation particles, negative determiners, indefinite pronouns, demonstrative pronouns, clitic pronouns, or negative pronouns.

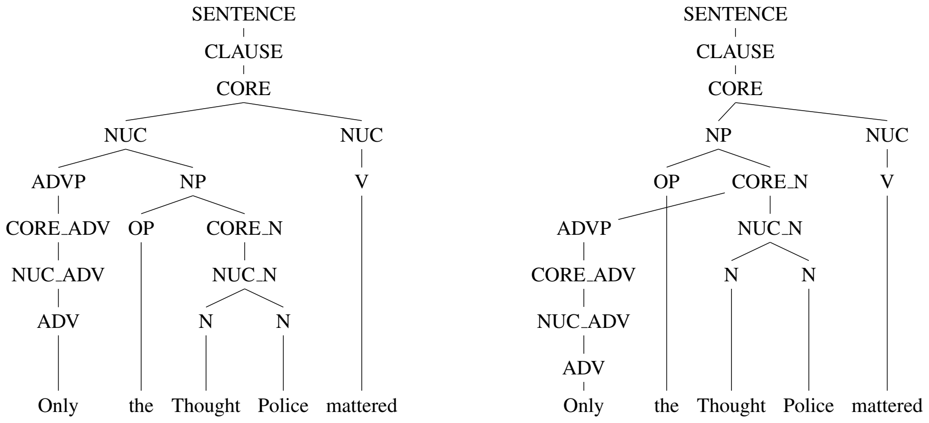
3 Impact on Annotation Effort

Correcting automatically pre-annotated data facilitates the annotation of the treebank because the data no longer need to be annotated from scratch. In this section we estimate the impact of pre-annotation on the human effort of creating treebank data. Specifically, we try to measure the mechanical effort it takes annotators to move, insert, delete, and relabel tree nodes in our graphical drag-and-drop annotation interface (Bladier et al., 2018). For this study, we ignore the cognitive cost of annotation decisions, which is much harder to measure.

Roughly speaking, the more similar a pre-annotated tree to the gold tree, the fewer drag-and-drop

[5] `https://gitlab.com/treegrasp/ud2rrg`

[6] The data is a snapshot of RRGparbank as of 2021-09-17.



(a) ud2rrg tree

(b) gold tree

Figure 4: Example pair of gold tree and corresponding ud2rrg output.

operations annotators will need. Established tree similarity measures include tree edit distance (TED) (Zhang and Shasha, 1989) and EVALB (Collins, 1997). However, it is well known that these measures tend to over-penalize attachment errors (Bangalore et al., 1998; Emms, 2008) because constituents that have to be reattached do not incur a unitary cost but rather a cost proportional to their size or to the length of the path between the predicted and the correct attachment site. This contrasts with our graphical annotation interface where reattachment is a single drag-and-drop operation. As an example, consider Figure 4. Here, we have to reattach one ADVP subtree and delete one NUC node in order to transform the ud2rrg tree into the gold tree. However, the tree edit distance is 6 because reattachment incurs a cost not only for the ADVP node but for all its descendants. Similarly, EVALB will count not 2 but 3 spans (NUC, NP, CORE_N) as "false positives" in the ud2rrg tree. This effect gets worse with longer reattachments.

We are not aware of a polynomial algorithm to compute optimal edit scripts between trees when reattachment is allowed as a single operation. Instead, we use an approximate but principled algorithm that counts the number of operations needed to turn the predicted tree into the gold tree when recreating the constituents of the gold tree by modifying the predicted tree in a strict bottom-up fashion, recreating smaller constituents first and then moving on to larger ones. This algorithm, called "bottom-up replugging" (BURP), is described in detail in Appendix B. In our example, BURP first recreates the CORE_N subtree, for which the ADVP subtree needs to be moved down (cost 1). It then recreates the NP subtree and deletes the NUC node (cost 1). The trees are now identical, with total cost 2, which is exactly the number of operations intuitively needed. While not necessarily optimal, we conjecture that BURP approximates the strategies that human annotators use to edit trees, and that its scores are therefore a better predictor of human annotation effort than TED or EVALB.

For our evaluation, we use all three measures. For TED and BURP, we normalize the score by the number of brackets in the gold RRG tree, since trees with a more complex internal structure require more editing than simpler trees. The results are given in Table 3.

4 Comparison with statistical parsing

We compare the output of ud2rrg with parsing the sentences using the statistical neural parser ParTAGe (Bladier et al., 2020), developed for RRG-based tree rewriting grammars. We evaluate how much training data is needed for the statistical parser to outperform the rule-based conversion approach. For the experiments, we did not distinguish between silver and gold data[7] but split all gold and silver data up into 4 385 training, 542 development, and 526 test sentences for English. We randomly shuffle the training data and use the first n trees for training. Our experiments show that the statistical parser needs around

[7] Silver sentences are annotated by one annotator whereas gold sentences are annotated by at least two annotators. We had 5 228 gold and 225 silver sentences in the English subcorpus in total.

approach	train sz.	failures		nTED		LF1 (exact match)		nBURP	
		dev	test	dev	test	dev	test	dev	test
ud2rrg		0	0	0.32	0.34	76.97 (90)	76.51 (84)	0.20	0.21
statist.	500	137	131	0.43	0.42	62.65 (70)	63.45 (85)	0.64	0.63
parser	1 000	0	1	0.35	0.35	68.93 (88)	70.27 (85)	0.30	0.29
	2 000	0	0	0.27	0.27	75.35 (128)	76.13 (113)	0.22	0.21
	3 000	0	0	0.25	0.24	77.93 (135)	78.73 (133)	0.19	0.18
	4 000	0	0	0.23	0.22	79.56 (149)	80.62 (135)	0.18	0.17
	>4 000	0	0	0.23	0.22	79.75 (157)	80.30 (137)	0.17	0.16
# sent.		542	526	542	526	542	526	542	526
∅ len.		13.99	14.02	13.99	14.02	13.99	14.02	13.99	14.02

Table 3: Comparison of UD parsing for English followed by rule-based ud2rrg conversion vs. statistical RRG parsing (Bladier et al., 2020), depending on the amount of RRG training data available. The evaluation does not consider function tags and punctuation. The numbers in brackets indicate the amount of exactly matched produced trees. Sentences that could not be converted/parsed are counted in the evaluation as flat dummy trees. We use our BURP measure (normalized by number of constituents in the gold tree) as well as tree edit distance (Zhang and Shasha, 1989) normalized in the same way, and EVALB LF1 (Collins, 1997). All three measures show that statistical parsing starts to outperform ud2rrg at around 2 000 training sentences.

2 000 pre-annotated trees for training to surpass the rule-based conversion.

We also evaluate the UD conversion on other languages (see Table 4).[8] In cases where ud2rrg could not convert UD parses to trees, we evaluate the scores as if the trees were annotated from scratch. Concretely, we measure the distance from flat dummy trees where each pre-terminal has a dummy POS tag and attaches directly to the root. The results show that about a fifth of the sentences are converted directly to the gold standard for different languages and in general the annotators' effort is reduced for the majority of sentences compared to annotation from scratch (represented as baseline in Table 4). These findings clearly show that using the rule-based UD conversion approach can be a good starting point for pre-annotation of a multilingual treebank.

language	baseline		ud2rrg		# sents	∅ len.	failures	# sents
	nBURP	LF1	nBURP	LF1	(annot.)	(annot.)		(entire corpus)
de	1.24	6.56	0.18	79.24 (926)	5723	17.00	9	6661
fr	1.22	8.97	0.21	79.80 (402)	2177	12.57	1	7261
ru	1.18	7.64	0.20	79.96 (939)	4635	11.76	0	6669
fa	1.16	9.14	0.30	72.09 (211)	1110	9.01	37	6604

Table 4: Comparison of normalized BURP and EVALB F1 scores of ud2rrg for German, French, Russian, and Farsi evaluated on the full set of annotated sentences without taking into account punctuation and function tags. The baseline is annotation from scratch, starting with flat dummy trees. For sentences where ud2rrg fails, we fall back to the baseline. The numbers in brackets show produced trees exactly matching with gold annotations.

5 Related Work

The availability of UD corpora for a big variety of languages makes them appealing to use for creating linguistic resources for different NLP tasks. Fancellu et al. (2017) and Reddy et al. (2017) describe algorithms for conversion of UD structures to logical forms enabling an almost language-independent transformation. Ranta and Kolachina (2017) develop an approach to convert UDs into abstract syntactic

[8] Note that these data do not fully reflect differences between languages in RRGparbank, since the annotation is still ongoing and the current amount of covered data and annotated syntactic phenomena is different for each language.

annotations to create treebanks based on the Grammatical Framework (GF) formalism for multilingual grammars (Ranta, 2011).

Closest to our work is that of Chiarcos and Fäth (2019) who define a RDF/SPARQL-based converter to RRG, using as input not only UD but also semantic role annotation. The data for which both the input (partially manually corrected UD) and the output is publicly available[9] consists of 351 examples from the textbook of Van Valin Jr. and LaPolla (1997). While their converter was developed on this kind of data, for us it presents a new domain. After normalizing away notational differences and ignoring operator attachment as well as POS tags (see below) but without any updates to ud2rrg, we obtained an nBURP of 0.16, an nTED of 0.18, and an F1 score of 85.75, with 15.38% exact matches. We then performed a qualitative comparison on 100 randomly chosen sentences to gain insights into types of mistakes and to inform future development. We summarize our findings here; the full results are provided in Appendix A.

Notational conventions A large part of the differences are purely notational and can be automatically normalized away: C&F attach all punctuation at the root whereas we leave it attached at smaller phrases as in UD. We mark *wh*-phrases with -REL or -WH labels, C&F don't. C&F mark arguments with ARG nodes, which we don't have, and peripheries with PERIPHERY nodes, while we mark the children with -PERI instead. Some nonterminals have slightly different names, e.g., COREn vs. CORE_N, LDP vs. PrDP, or ADJ vs. AP. We also ignore part-of-speech tags because C&F do not attempt to convert the input POS tags into the POS tags conventionally used in RRG analyses.

Tense, modality, and aspect operator attachment In RRG, tense operators attach at the CLAUSE level, modal operators at the CORE level, and aspect operators at the NUC level. This means that, e.g., a tensed auxiliary verb as in *she has seen* or a tensed modal verb as in *he can see* attach at more than one level, namely at both CLAUSE and NUC, and at both CLAUSE and CORE, respectively. In C&F's annotation, this is so. By contrast, our guidelines limit annotations to trees for ease of processing and by convention only attach at one level, which is typically CLAUSE. We tried to ignore these differences in attachment by removing the non-CLAUSE additional edges from C&F's annotation. 10 differences due to operator attachment remain. Since auxiliary and modal verbs form a closed class, the multiple attachment would be easy to restore.

Theoretical assumptions (51 instances) Some of the remaining differences can be explained by ud2rrg following conventions set down in our RRGbank-based annotation guidelines which differ from those followed in C&F's data. These are not mere notational differences but have potential theoretical significance because they reflect different assumptions about the internal structure of phrases, etc. These differences will be used to check and revise our annotation guidelines. For example, we annotate numerals using "quantifier phrases" (QPs), attach attributive APs at CORE_N rather than NUC_N level, assume a full AP rather than a simple nucleus for predicative adjectives, treat possessive pronouns like determiners and do not place them under NPIP, treat prepositions introducing adverbial clauses as clause linkage markers (CLM) rather than prepositions, do not distinguish CONJ from CLM, etc.

Bugs (25 instances) Some differences are bugs in ud2rrg which can easily be fixed in future development, e.g., failure to convert prepositions `mark`ing clauses into PPs rather than CLM-marked clauses, failure to handle `nmod:tmod` dependents as adverbial modifiers, attachment of *wh*-PPs in PrDP rather than PrCS, or failure to recognize *wh*-movement when the subject is a passive subject.

Limitations (84 instances) Telling the differences between an argument PP and a peripheral (adjunct) PP is hard and currently out of scope for ud2rrg. C&F use semantic role information to predict this. Relatedly, ud2rrg currently does not distinguish between PPs with and without internal layers (CORE_P and NUC_P).

Clause linkage (20 instances) Similarly, mapping `conj`, `ccomp`, and `xcomp` dependencies to the appropriate juncture-nexus type for linking clauses together is complex. As described in Section 2, we

[9] `https://github.com/acoli-repo/RRG`

have some rules to address this heuristically, but many cases are not yet covered and may also require semantic role or other lexical information to resolve.

Bad input (12 instances) Incorrect UD input sometimes leads to incorrect ud2rrg output. For example, the input data contain a number of unspecific `dep` relations which are then not correctly handled. There are also instances of wrongly resolved PP attachment ambiguity and the occasional confusion of a relative clause with an adverbial clause, and vice versa.

Error in gold standard (7 instances) Finally, we also discovered a handful of apparent errors in C&F's annotation. For example, the genitive suffix *'s* is always attached to the root instead of inside the NPIP, and some PrCS nodes appear to be spurious.

6 Conclusions

We have presented a rule-based algorithm for converting Universal Dependencies to RRG trees as a way to bootstrap RRG treebanks. By mapping UD nodes to RRG fragments and grammatical relations to operations that combine these fragments, it provides a principled mapping between the two formalisms. Language-independent at core, the algorithm can be extended with language-specific rules to incorporate lexical and other language-specific knowledge. We have shown that by basing RRG annotation on automatically converted trees, the number of tree manipulation operations that annotators have to perform is considerably reduced compared to annotating from scratch. We have also shown that for annotating English, a statistical parser trained on sentences annotated so far starts to produce more accurate trees than our rule-based conversion at around 2 000 training sentences. Finally, we have performed a detailed qualitative comparison with the output of another converter and pinned down the remaining issues for ours. In future work, we will consider applying the ud2rrg algorithm to the data from Parallel Universal Dependencies corpora (Zeman et al., 2020). Moreover, ud2rrg allows bootstrapping of further RRG treebanks for different languages, based on existing UD treebanks.

Acknowledgments

We thank two anonymous reviewers for their insightful comments, as well as all people involved in the creation and annotation of RRGparbank (David Arps, Davy Baardink, Kata Balogh, Elif Benli, Sarah Fabian, Valeria Generalova, Zahra Ghane, Robin Möllemann, Natalia Moors, Rainer Osswald) for their help with collecting the data and fruitful discussions. We also thank Behrang QasemiZadeh, Jule Pohlmann and Roland Eibers for preparing the German data, Rainer Osswald for integrating the French data and completing the German data, Naima Grebe for helping with the implementation of special conversion rules for English phase verbs, and Robert D. Van Valin Jr. for contributing to annotation decisions. This work was carried out as a part of the research project TreeGraSP[10] funded by a Consolidator Grant of the European Research Council (ERC).

References

Srinivas Bangalore, Anoop Sarkar, Christy Doran, and Beth-Ann Hockey. 1998. Grammar and parser evaluation in the XTAG project. In *Proceedings of LREC Workshop on Evaluation of Parsing Systems*.

Tatiana Bladier, Andreas van Cranenburgh, Kilian Evang, Laura Kallmeyer, Robin Möllemann, and Rainer Osswald. 2018. RRGbank: a role and reference grammar corpus of syntactic structures extracted from the penn treebank. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)*. Linköping Electronic Conference Proceedings.

Tatiana Bladier, Jakub Waszczuk, and Laura Kallmeyer. 2020. Statistical parsing of tree wrapping grammars. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6759–6766, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

[10] `https://treegrasp.phil.hhu.de`

Christian Chiarcos and Christian Fäth. 2019. Graph-Based Annotation Engineering: Towards a Gold Corpus for Role and Reference Grammar. In Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski, editors, *2nd Conference on Language, Data and Knowledge (LDK 2019)*, volume 70 of *OpenAccess Series in Informatics (OASIcs)*, pages 9:1–9:11, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain, July. 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, 07.

Martin Emms. 2008. Tree-distance and some other variants of evalb. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco.

Tomaž Erjavec. 2017. MULTEXT-East. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 441–462, Dordrecht. Springer Netherlands.

Federico Fancellu, Siva Reddy, Adam Lopez, and Bonnie Webber. 2017. Universal dependencies to logical forms with negation scope. *arXiv preprint arXiv:1702.03305*.

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

Birgit Hamp and Helmut Feldweg. 1997. GermaNet - a lexical-semantic net for German. In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

Aravind K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, pages 69–123. Springer, Berlin.

Karin Kipper-Schuler. 2005. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia, May. European Language Resources Association (ELRA).

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 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May. European Language Resources Association.

George Orwell. 1972. *1984*. Gallimard. French translation by Amélie Audiberti (first published 1950).

George Orwell. 2003. *1984*. Ullstein, 37th edition. German translation by Kurt Wagenseil (first published 1950 by Alfons Bürger Verlag).

Rainer Osswald and Laura Kallmeyer. 2018. Towards a formalization of Role and Reference Grammar. In Rolf Kailuweit, Eva Staudinger, and Lisann Künkel, editors, *Applying and Expanding Role and Reference Grammar*, (NIHIN Studies), pages 355–378. Albert-Ludwigs-Universität, Universitätsbibliothek, Freiburg.

Aarne Ranta and Prasanth Kolachina. 2017. From universal dependencies to abstract syntax. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 107–116.

Aarne Ranta. 2011. *Grammatical framework: Programming with multilingual grammars*, volume 173. CSLI Publications, Center for the Study of Language and Information Stanford.

Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. *arXiv preprint arXiv:1702.03196*.

Brian Roark. 2002. Evaluating parser accuracy using edit distance. In *Proceedings of the LREC 2002 workshop: Beyond PARSEVAL: Towards Improved Evaluation Measures for Parsing Systems*.

Geoffrey Sampson and Anna Babarczy. 2003. A test of the leaf-ancestor metric for parse accuracy. *Journal of Natural Language Engineering*, 9:365–380.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium, October. Association for Computational Linguistics.

Robert D. Van Valin Jr. and Randy J. LaPolla. 1997. *Syntax: Structure, meaning and function*. Cambridge University Press.

Robert D. Van Valin Jr. 2005. *Exploring the Syntax-Semantics Interface*. Cambridge University Press.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Noëmi Aepli, Željko Agic, Lars Ahrenberg, et al. 2020. Universal dependencies 2.5. *LINDAT/CLARIAHCZ digital library at the Institute of Formal and Applied Linguistics (UFAL), Faculty of Mathematics and Physics, Charles University. url: http://hdl. handle. net/11234/1-3226*.

Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262.

A Qualitative Comparison with Chiarcos and Fäth (2019)

The following table contains the results of the qualitative evaluation of our converter on 100 randomly selected example sentences from Van Valin Jr. and LaPolla (1997) as annotated by Chiarcos and Fäth (2019). The first column indicates the sentence number in their release, the second the type of difference (different **op**erator attachment, **theo**retical assumption, **bug**, **limit**ation, clause **link**age, bad input (**ud**), or error in **gold** standard), and the third column contains a brief description of the difference. Empty second and third columns indicates sentences from our sample with no difference after normalization.

sentence	type	description
5	op	neg at CLAUSE vs. CORE
5	bug	failure to attach fronted non-wh object in PrCS
8	ud	dobj → dep
8	limit theo	PP internal structure
9	bug	fronted wh-PP attached to PrDP instead of PrCS
9	limit theo	PP internal structure
12	limit	failure to recognize PERI
17	limit theo	PP internal structure
17	ud	fronted wh-object attached with dep
18	ud	nmod:tmod → dep
18	limit	failure to recognize PERI
18	limit theo	PP internal structure
24		
26	limit theo	PP internal structure
29	limit theo	PP internal structure
30	limit theo	PP internal structure
31	limit theo	PP internal structure
31	limit theo	PP internal structure
35	acoli	”that” treated as NP when it is a determiner
48	acoli	’s attached to root
48	theo	AP-PERI always attaches at CORE_N, not NUC_N
48	limit theo	PP internal structure
50	limit theo	PP internal structure
54	limit theo	PP internal structure
54	theo	ADVP-PERI always attaches at CORE, not NUC
54	theo	AP-PERI always attaches at CORE, not NUC
55	theo	ADVP-PERI always attaches at CORE, not NUC
55	theo	AP-PERI always attaches at CORE, not NUC
55	bug	failure to recognize PoDP
55	limit theo	PP internal structure
58	theo	”by” passive subject treated as argument, not adjunct
58	limit theo	PP internal structure
59	limit theo	PP internal structure
62		
67	theo	”by” passive subject treated as argument, not adjunct
68	limit theo	PP internal structure
71	limit	failure to recognize PERI
74	ud	compound → dobj
74	limit	failure to recognize PERI
76	theo	possessive pronoun treated as definiteness operators, not placed in NPIP
76	theo	AP-PERI always attaches at CORE_N, not NUC_N
76	limit theo	PP internal structure

89		
91		
99	bug	raising construction where the subordinated predicate is an adjective is wrongly classified as dependency parsing error
103	ud	acl:relcl → advcl
103	ud	”to whom” is not a subtree
103	limit theo	PP internal structure
104	limit theo	PP internal structure
104	limit theo	PP internal structure
104	bug	failure to recognize PrCS when subject is marked nsubjpass
105	limit theo	PP internal structure
105	limit theo	PP internal structure
105	bug	failure to recognize PrCS when subject is marked nsubjpass
111		
112		
120	limit	failure to recognize PERI
120	limit theo	PP internal structure
120	limit	failure to recognize PERI
123	limit theo	PP internal structure
123	limit	failure to recognize PERI
123	limit	failure to recognize PERI
125	limit	failure to recognize PERI
125	limit	failure to recognize PERI
127		
132	theo	to-infinitive that replaces a relative clause treated as CLAUSE-PERI and attached in NUC_N instead of CORE attached at CORE_N
132	limit theo	PP internal structure
133	limit theo	PP internal structure
134	ud	NP with relcl instead of SENTENCE with fronted dobj
135	limit theo	PP internal structure
139		
140	limit theo	PP internal structure
141	limit theo	PP internal structure
146	limit theo	PP internal structure
150	limit theo	PP internal structure
153	theo	AP-PERI always attaches at CORE_N, not NUC_N
153	limit theo	PP internal structure
153	limit theo	PP internal structure
153	limit theo	PP internal structure
153	limit theo	PP internal structure
153	limit theo	PP internal structure
159	limit theo	PP internal structure
159	limit theo	PP internal structure
160	limit theo	PP internal structure
160	limit theo	PP internal structure
162	limit theo	PP internal structure
168	limit theo	PP internal structure
168	limit	failure to recognize PERI
171	limit	failure to recognize PERI
171	limit theo	PP internal structure
172	limit theo	PP internal structure

172	limit	failure to recognize PERI
174	limit theo	PP internal structure
182		
183	limit	failure to recognize PERI
183	limit theo	PP internal structure
188	bug	fronted wh-PP attached to PrDP instead of PrCS
188	limit theo	PP internal structure
190	op	neg at CLAUSE vs. CORE
190	link	parataxis handled as CORE cosubordination instead of SENTENCE coordination
190	limit	failure to recognize/handle cleft construction
200	theo	single complex NUC instead of NUC cosubordination
200	theo	predicative adjective: simple NUC vs. AP
200	link	CORE coordination vs. cosubordination
202	link	CORE coordination vs. cosubordination
205	link	CORE coordination vs. subordination
205	theo	predicative adjective: simple NUC vs. AP
207		
211	theo	single complex NUC instead of NUC cosubordination
211	theo	predicative adjective: simple NUC vs. AP
217	bug	failure to handle nmod:tmod as adverbial
217	theo	single complex NUC instead of NUC cosubordination
217	theo	predicative adjective: simple NUC vs. AP
225	link	CORE coordination vs. CLAUSE subordination
229	bug	advcl as PP-PERI vs. CLAUSE
229	limit theo	PP internal structure
229	limit theo	PP internal structure
231	op	modal verbs attach at CORE
233	op	modal verbs attach at CORE
233	link	CORE coordination vs. cosubordination
236	limit	failure to recognize PERI
236	limit theo	PP internal structure
237		
239	link	CLAUSE vs. CORE subordination
239	acoli	spurious PrCS?
240	ud	nominalized clause parsed wrong
240	theo	”by” passive subject treated as argument, not adjunct
247	limit theo	PP internal structure
247	bug	failure to handle nmod:tmod as adverbial
247	link	CLAUSE vs. CORE subordination
248	bug	advcl as PP-PERI vs. CLAUSE
248	limit theo	PP internal structure
255	ud	xcomp → dep
258	link	CORE coordination vs. cosubordination
259	link	CORE coordination vs. subordination
260	link	CORE coordination vs. cosubordination
260	theo	possessive pronoun treated as definiteness operators, not placed in NPIP
260	theo	possessive pronoun treated as definiteness operators, not placed in NPIP
260	bug	advcl as PP-PERI vs. CLAUSE
261	link	CORE coordination vs. cosubordination
261	theo	possessive pronoun treated as definiteness operators, not placed in NPIP

261	theo	possessive pronoun treated as definiteness operators, not placed in NPIP
261	bug	advcl as PP-PERI vs. CLAUSE
263	link	CORE coordination vs. subordination
265	link	CLAUSE coordination vs. cosubordination
265	bug	advcl as PP-PERI vs. CLAUSE
265	theo	possessive pronoun treated as definiteness operators, not placed in NPIP
265	limit theo	PP internal structure
276	theo	”by” passive subject treated as argument, not adjunct
276	theo	”by” passive subject attached at higher vs. lower CORE
278	bug	fronted advcl not in PrDP
278	bug	ud2rrg advcl as PP vs. CLAUSE
292	link	CORE subordination vs. cosubordination
293		
294	limit theo	PP internal structure
294	link	CLAUSE in CORE vs. CORE subordination
294	acoli	spurious PrCS?
294	theo	predicative adjective: simple NUC vs. AP
295	limit theo	PP internal structure
295	link	CLAUSE under CORE vs. CORE subordination
297	limit theo	PP internal structure
297	acoli	’s attached to root
297	acoli	’s attached to root
297	theo	QP
297	theo	QP
297	theo	AP-PERI always attaches at CORE, not NUC
297	theo	AP-PERI always attaches at CORE, not NUC
297	theo	we don’t distinguish between CONJ and CLM
298	theo	we don’t distinguish between CONJ and CLM
298	theo	QP
298	theo	QP
298	limit	article of coordinated NP attaches too low
301	theo	QP
301	theo	AP-PERI always attaches at CORE, not NUC
305	op	at CLAUSE vs. CORE
305	limit theo	PP internal structure
305	limit theo	PP internal structure
305	ud	PP attachment
305	limit	failure to recognize PERI
305	op	modal verbs attach at CORE
305	link	NP-CLAUSE subordination vs. CORE subordination
309	theo	neg at CLAUSE vs. CORE
309	limit theo	PP internal structure
309	ud	advcl → acl:relcl
309	op	modal verbs attach at CORE
309	limit	failure to recognize PERI
312	op	modal verbs attach at CORE
312	bug	neg determiner attached wrongly
312	limit	failure to recognize PERI
318	link	CORE coordination vs. CORE cosubordination
318	limit	failure to recognize PERI
324	theo	QP

324	theo	AP-PERI always attaches at CORE_N, not NUC_N
324	ud	discontinuous wh-PP (stranding) not parsed correctly
324	bug	failure to handle nmod:tmod as adverbial
324	theo	relative clause attaches at NUC_N, not CORE_N
325	theo	QP
325	theo	QP
325	theo	AP-PERI always attaches at CORE_N, not NUC_N
325	theo	AP-PERI always attaches at CORE_N, not NUC_N
325	limit	PP attaches too low in coordinated NP
325	theo	we don't distinguish between CONJ and CLM
326	limit theo	PP internal structure
326	op	modal verbs attach at CORE
326	theo	we don't distinguish between CONJ and CLM
326	bug	failure to handle elliptical conjunct
326	bug	failure to recognize PoDP
331	theo	single complex NUC instead of NUC cosubordination
331	acoli	's attached to root
336	bug	untensed auxiliary treated as OP-TNS
337	bug	failure to handle nmod:tmod as adverbial
348		
350		
350	bug	advcl as PP-PERI vs. CLAUSE
350	limit	failure to recognize PERI
350	theo	we don't distinguish between CONJ and CLM
350	bug	NP conjunction attaches to PP
350	bug	failure to recongize PrDP
350	link	CORE cosubordination vs. CORE under NUC
350	op	modal verbs attach at CORE
350	theo	AP-PERI always attaches at CORE_N, not NUC_N
350	theo	predicative adjective: simple NUC vs. AP
350	limit theo	PP internal structure

B Computing Tree Distance Using Bottom-up Replugging (BURP)

We describe BURP ("bottom-up replugging"), an algorithm that computes an edit script between two trees with identical spans, such as two different natural-language constituent parse trees over the same sentence. Potential applications include evaluating the performance of constituent parsers and estimating the annotator effort in a semi-automatic annotation scenario.

Similar metrics include tree-distance (Zhang and Shasha, 1989; Emms, 2008), EVALB (Collins, 1997), string-distance applied to tree linearizations (Roark, 2002), and the leaf-ancestor metric (Sampson and Babarczy, 2003). None of them explicitly models the possibility of re-attaching a subtree to a different node, and they thus tend to over-penalize attachment errors as every constituent containing a moved subtree is affected (Bangalore et al., 1998). Although Roark (2002) and Emms (2008) propose strategies that mitigate this, subtree re-attachment is still handled as pair of delete and insert operations, thus its cost cannot be freely chosen but is necessarily the sum of the two.

BURP differs from all these algorithms by trying to explicitly simulate the way human annotators using graphical annotation interface correct trees. We assume the following basic operations to be availalbe: relabeling a node, deleting an internal node (implicitly reattaching all its children to its parent), inserting a node below another node (so that children of the existing node become children of the new node), and moving a node (that has at least one sibling) to a different parent. Given these operations, one question to ask is what is the optimal set of operations to transform the source (or predicted) tree into the target (or gold) tree, given some cost for each operation (in the following, we assume that every operation has cost

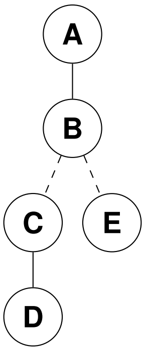


Figure 5: An example tree consisting of three maximal unary chains. Dashed edges indicate the boundaries between chains.

1, but they can easily be weighted differently). This is an NP-complete problem. Another, and perhaps more interesting question is how many operations human annotators need. We conjecture that BURP mimicks the human annotation process to some degree, and thus gives script lengths that correlate better with human annotator effort than other measures.

Sketch of the algorithm BURP transforms the source tree into the target tree in a bottom-up fashion, recreating smaller subtrees of the target tree in the source tree first and then moving on to larger ones until the root is reached and the whole tree transformed. To this end, the target tree is first divided up into maximal unary *chains*, as illustrated in Figure 5. To simplify the description, we will often refer to a chain as if all its nodes have been contracted into one, i.e. we say that in the example, chain AB has two children CD and E. We will also occasionally refer to a subtree and its root as the same entity if the context makes it clear. BURP does a post-order traversal of the chains in the target tree and at each chain transforms a part of the source tree into the subtree under that chain. All children of the chain have at this point already been recreated, and they are re-used, even if this is not guaranteed to give an optimal edit script. This is how BURP cuts the NP-complete problem down to a polynomial one. The local decisions, *viz.*, which part of the source tree to transform into the current target chain, however, are optimized for minimal local cost.

Definitions The *span* (or *yield*) of a tree is the set of its leaves. Note that we do not assume spans to be contiguous.

Inputs The inputs to BURP are a source tree T_1 and a target tree T_2 with identical spans.

Data Structures and Initialization While transforming T_1 into T_2, we will temporarily remove subtrees from T_1 and thus take it apart into multiple parts. We maintain a set P of these parts, which we initialize as $P := \{T_1\}$. We say that a node is "free" if it is the root of a tree in P.

The Traversal We do a post-order traversal of the chains in T_2, recreating the subtrees under them in as subtrees of trees in P. Thus, when we visit a chain, all of its children have already been recreated. Let p_2 be the currently visited target chain and C its recreated children in the trees in P. In the example in Figure 6, $p_2 = \mathsf{BDE}$ and $C = \{\mathsf{F}, \mathsf{G}, \mathsf{H}, \mathsf{I}, \mathsf{J}\}$. We then pick an *extended source chain* or *x-chain*, i.e., a path p_1 in some tree in P such that $p_1 = n_1 n_2 \dots n_N$ with $N \geq 1, n_N \in C$. In the example, $p_1 = \mathsf{ABCF}$. The subtree under p_1 is then deterministically transformed into that under p_2 with the minimal number of operations and assuming that the subtrees under all $c \in C$ remain unchanged.[11] The transformation consists of the following steps:

[11] The reason for including one of the transformed children in p_1 is to allow for the case where the rest of the source chain is empty and all nodes in p_2 have to be inserted during the transformation. An empty source chain would not specify where to insert these nodes. Note also that $n_1 \dots n_{N-1}$ need not be a unary chain (in the example, B has another child P); it will be transformed into one.

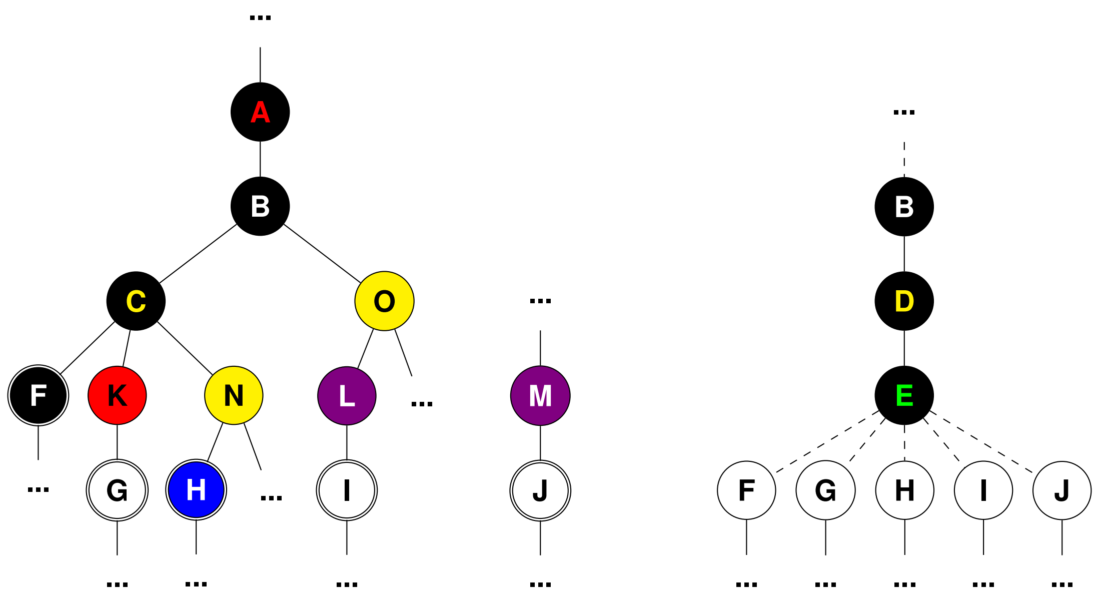


Figure 6: Example state of the algorithm, with two source tree fragments on the left and a target tree fragment on the right. Black nodes are parts of the target chain or source x-chain. Doubly circled nodes are already-recreated children in the source tree fragments. Yellow nodes will be freed, blue ones moved, red ones pruned, and violet ones moved and pruned.

1. **Move down.** For each topmost descendant of $n_1 \dots n_{N-2}$ whose span is a subset of the target span (meaning the span of p_2) but which is not dominated by n_{N-1}, move it to n_{N-1}. In the example, L is moved to C (cost 1).

2. **Free above.** For each child of $n_1 \dots n_{N-2}$ that is not in p_1, "free" it, i.e., remove it and add it to P as a free subtree. It will find its place is the transformed tree later. In the example, O is freed (cost 1).

3. **Edit chain.** Insert, delete, and relabel nodes in $n_1 \dots n_{N-1}$ so as to make the chain identical to p_2. The cost is the Levenshtein distance (Levenshtein, 1966) between both sequences.[12] In our example, A is deleted, C is relabeled D, and E is inserted (cost 3).

4. **Move up.** For each topmost descendant of n_{N-1} whose span is a subset of the target span but which is not a child of n_{N-1}, move it to n_{N-1}. In the example, H is moved to C (cost 1).

5. **Free below.** For each child of n_{N-1} whose span is not a subset of the target span, free it. In our example, N is freed (cost 1).

6. **Move in.** For each topmost node whose span is a subset of the target span and that is not yet a child of n_{N-1}, move it there. If that node is a root, there is no cost because the cost of moving was already incurred when the node was freed.[13] In the example, M is moved in (cost 1).

7. **Prune.** For every node between n_{N-1} and any $c \in C$, delete it. In the example, K, L, and M are deleted (cost 3).

Postcondition After visiting the root of T_2, P contains exactly one tree, which is identical to T_2.

[12] Our graphical annotation interface does not currently allow for inserting a node directly above another node in a unary chain if the latter has siblings. This could be taken into account by disallowing insertions at the beginning of the Levenshtein edit script.

[13] The cost is incurred early, by the freeing operation, not by the subsequent "moving in", so it can be attributed to the x-chain that necessitates the moving. Annotators often do not have a place where they can put removed subtrees temporarily; we assume that they will immediately move the subtree to the node where it will eventually end up.

Search For every visited target chain, we pick an x-chain that minimizes the local cost. The final edit chain and cost still depends on how ties between locally optimal x-chains are broken, and on the exact order of traversal. In our current implementation[14], the leaves are assumed to be ordered (as the words are in natural language), and the postorder traversal proceeds from left to right. Ties between x-chains are currently broken by preferring chains that are in more recently freed subtrees, further to the right in the tree, and longer. A closer approximation to the optimal edit script could be achieved, e.g., by randomizing this and doing multiple restarts.

[14] `https://github.com/texttheater/burp`

For the Purpose of Curry: A UD Treebank for Ashokan Prakrit

Adam Farris*
San Mateo High School
adamfarris@gmail.com

Aryaman Arora*
Georgetown University
aa2190@georgetown.edu

Abstract

We present the first linguistically annotated treebank of Ashokan Prakrit, an early Middle Indo-Aryan dialect continuum attested through Emperor Ashoka Maurya's 3rd century BCE rock and pillar edicts. For annotation, we used the multilingual Universal Dependencies (UD) formalism, following recent UD work on Sanskrit and other Indo-Aryan languages. We touch on some interesting linguistic features that posed issues in annotation: regnal names and other nominal compounds, "proto-ergative" participial constructions, and possible grammaticalizations evidenced by *sandhi* (phonological assimilation across morpheme boundaries). Eventually, we plan for a complete annotation of all attested Ashokan texts, towards the larger goals of improving UD coverage of different diachronic stages of Indo-Aryan and studying language change in Indo-Aryan using computational methods.

1 Introduction

Ashokan Prakrit is the earliest attested stage and among the most conservative known forms of Middle Indo-Aryan (MIA), represented by inscriptions in the form of rock and pillar edicts commissioned by the Mauryan emperor Ashoka (*aśōka*[1]) in the 3rd century BCE. The Indo-Aryan languages are the predominant language family in the northern (and insular southern) parts of the Indian subcontinent, and consitute a branch of the widespread Indo-European family. They are generally divided into three historical stages: Old Indo-Aryan (OIA; Sanskrit, both the language of Vedic and of later Classical texts, as well as unattested varieties suggested by dialectal variation in later stages), Middle Indo-Aryan (MIA; Ashokan Prakrit, Pali, the Dramatic Prakrits, and early koinés of the Hindi Belt), and New Indo-Aryan (NIA; modern Indo-Aryan languages such as Hindi–Urdu, Assamese, Marathi, Dhivehi, Kashmiri, Khowar, etc.).

Diachronically, Ashokan Prakrit is a descendant of Old Indo-Aryan varieties (some of which are attested through Vedic and Classical Sanskrit) and is a precursor to regional fragmentation of Middle Indo-Aryan into Pali, the Dramatic Prakrits, and eventually the NIA languages. Ashokan Prakrit is a dialect continuum rather than a standardized language, but the three dialect zones are not divergent enough to prove mutually unintelligible (Oberlies, 2003).

Universal Dependencies (Nivre et al., 2016; de Marneffe et al., 2021) is a multilingual formalism for treebanking, including annotation guidelines for dependency relations, morphological analysis, part-of-speech tagging, and other linguistic features. Several New Indo-Aryan languages (Bhatt et al., 2009;

[*] Equal contribution.

[0] The example of *sūpāt^{h}āya* 'for the purpose of curry' (discussed further in §5.3) inspired the title of this paper.

Glossing abbreviations: 1 = first person, 3 = third person, ACC = accusative, ALTER = alterphoric, CAUS = causative, DAT = dative, DEM = demonstrative, EMPH = emphatic particle, F = feminine, GEN = genitive, IND = indicative, INS = instrumental, LOC = locative, M = masculine, N = neuter, NOM = nominative, PASS = passive, PL = plural, PPP = past passive participle, PRS = present, PST = past, SG = singular.

[1] Throughout this work, we use a newly devised transliteration scheme, devised by Samopriya Basu, based on the International Alphabet of Sanskrit Transliteration (IAST) which is standard in Indological work, as well as influences from the IPA and Americanist systems. Divergences from IAST are: 1. indication of aspiration and breathy voice with superscript ⟨^{h}⟩, 2. explicit marking of ⟨ē⟩ and ⟨ō⟩ as long vowels, 3. overdot for visarga ⟨ḣ⟩ and anusvara ⟨ṁ⟩, instead of the underdot, to avoid confusion with retroflexion.

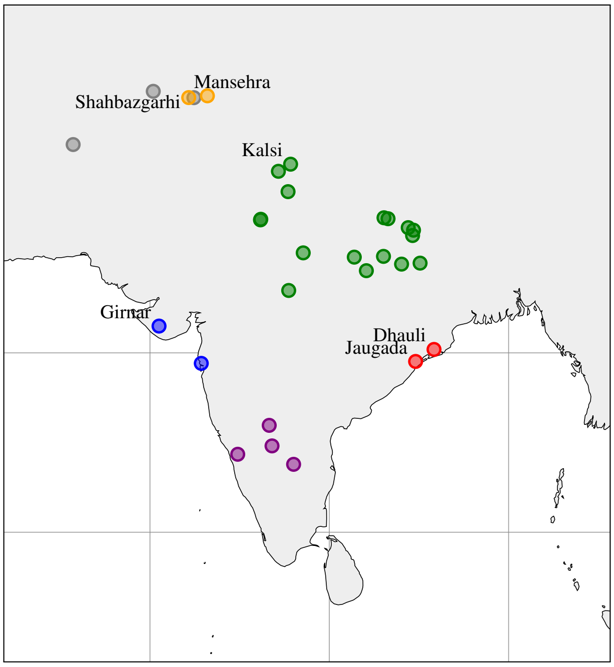


Figure 1: Locations of the various Ashokan inscriptions and edicts in the Indian Subcontinent, coloured by their usual geographic grouping (not by linguistic isoglosses). Points in grey in the northwest are inscriptions that are not in Ashokan Prakrit (instead, Aramaic and Greek).

Tandon et al., 2016; Ravishankar, 2017) and Sanskrit (Kulkarni et al., 2020; Hellwig et al., 2020; Dwivedi and Zeman, 2017) have treebanks annotated using UD or other syntactic formalisms, but thus far there is no treebank for a MIA language, leaving a gap in Indo-Aryan historical corpora. Within MIA, Ashokan Prakrit has an unusual corpus of parallel texts representing multiple geographical dialects, conducive to the study of Indo-Aryan linguistic fragmentation using computational tools.

To this end, we began UD annotation of a digitized Ashokan Prakrit corpus under the **Digitizing Imperial Prakrit Inscriptions** (DIPI)[2] project. We will present some interesting annotation issues that arose, both in the context of Indo-Aryan comparative linguistics and for the Universal Dependencies annotation scheme, and suggest future directions for historical and dialectological corpus linguistic work in the Indo-Aryan family.

2 Related work

The first Ashokan edicts were deciphered by James Prinsep in the 1830s (Kopf, 1969). Since then, they have played an important role in the historical study of Ashoka and the Mauryan Empire, sociological and religious study on early Buddhism and other heterodox Dharmic sects (Smith et al., 2016; Scott, 1985), and, of course, linguistic work from a historical and social perspective. Figure 1 shows the locations of the known Ashokan inscriptions, with labels on the locations particularly relevant to this paper.

There are several works which attempt a broad comparative study of the inscriptions with reference to Sanskrit (Woolner, 1924; Hultzsch, 1925; Mehendale, 1948; Bloch, 1950; Sen, 1956; Oberlies, 2003). Like most historical linguistic work on IA, these works focus mostly on phonology and, to a lesser extent, morphology to the detriment of syntax and semantics (Varma, 1947).

On the computational side, the only digitized and machine-readable version of the Ashokan edicts is the Ashoka Library (Braarvig et al., 2014), which is sourced from Hultzsch (1925) and thus missing more recently discovered inscriptions.

Other UD corpora and their annotation guidelines were also helpful to our own annotation process, e.g. Scarlata et al. (2020). Hand-prepared Ashokan Prakrit inflectional tables based on data harvested from Mehendale (1948) were of use, in addition to Sanskrit dictionaries (Monier-Williams, 1899; Sircar,

[2] From Shahbazgarhi, Mansehra *dipi* ‘rescript, writing’, as opposed to the lateralized variant *lipi* attested in other dialects.

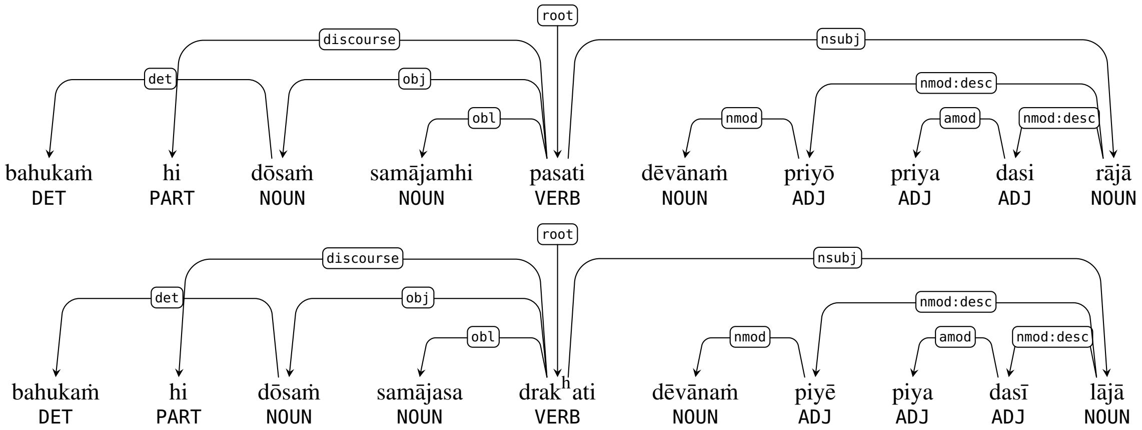


Figure 2: Dependency parse of the fourth sentence of Major Rock Edict 1 as found in two locations. The top is from Girnar, representing the Western dialect, and the bottom is from Jaugada, representing the Eastern dialect.

1966) and morphological analysers (Huet, 2005).[3]

3 Corpus

The Ashokan Prakrit texts available to us constitute a very limited corpus. They are royal inscriptions concerning the promotion of Buddhist morality, administration of the Mauryan Empire, and records of Ashoka's magnanimous deeds (such as his conversion to Buddhism). They directly address the public, and all evidence points to Ashokan Prakrit being a semi-standardized but still fairly accurate reflection of vernacular language, given the geographical dialect variation and communicative purpose of the texts.

We began with transcribed edicts from the Ashoka Library (Braarvig et al., 2014). Annotation began in June 2021 and was done in Google Sheets simultaneously by two linguistically-informed annotators with discussions to resolve disagreements. Although Google Sheets is not the conventional choice of tool for such a project, existing UD annotation tools were found to be lacking a convenient means of editing `FEATS` columns in a CONLLU file, as well as supplying additional columns (e.g. etymologies). Additionally, this allowed us to avoid setting up the server required for collaborative annotation with tools like UD Annotatrix (Tyers et al., 2017) A guidelines document was added to as the analysis of tricky constructions was decided upon.

Given the parallel nature of the corpus, annotations for a particular edict at one location could be transferred with little modification to that of another location. An example of this is given in figure 2, which only shows POS-tag and dependency parse UD annotations of a parallel sentence, glossed below.

(1) bahukaṁ hi dōsaṁ samājamhi pasati Dēvānaṁ-
very EMPH evil:ACC.M.SG meeting:LOC.M.SG see:PRS.IND.3.SG god:NOM.M.PL
priyō Priya- dasi rājā
beloved:NOM.M.SG kindly looking:NOM.M.SG king:NOM.M.SG
'King Beloved-of-the-Gods Looking-Kindly sees much evil in festival meetings.' (Girnar 1:4)

Thus, we used the well-preserved edicts at Girnar as the main annotation document, and annotated other editions only after finalising the corresponding Girnar version. Table 1a gives statistics about the annotated corpus.

4 Annotation and analysis

We annotated using the standard Universal POS tag inventory and Universal Dependency Relations from Universal Dependencies v2, with some additional dependency subtypes: `acl:relcl`, `advmod:lmod`, `advmod:tmod`, `advmod:neg`, `nmod:desc` (discussed in §5.1.1), `obl:lmod`, `obl:tmod`. Overall UPOS counts are given in table 2.

[3] The Sanskrit Grammarian (Huet, 2005) has a web interface at `https://sanskrit.inria.fr/DICO/grammar.html`.

	Doc.	Sent.	Tok.
Girnar	5	43	534
Shahbazgarhi	3	14	158
Mansehra	1	8	87
Kalsi	1	8	85
Jaugada	1	8	89
Dhauli	1	3	20
Total	12	84	973

(a) DIPI corpus composition, grouped by source location of the annotated inscriptions.

Feature	Measure	Val.
`UPOS`	Cohen's κ	0.949
`HEAD`	UAS	0.857
`DEPREL`	Label score	0.776
`HEAD+DEPREL`	LAS	0.673

(b) Agreement scores between two annotators on Girnar Major Rock Edict 7.

Table 1: Metrics about the DIPI corpus.

UPOS	Count	%
`NOUN`	345	35.5%
`ADJ`	136	14.0%
`VERB`	106	10.9%
`ADV`	83	8.5%
`CCONJ`	78	8.0%
`PRON`	47	4.8%
`PART`	42	4.3%

UPOS	Count	%
`DET`	42	4.3%
`NUM`	35	3.6%
`PROPN`	22	2.3%
`X`	14	1.4%
`SCONJ`	11	1.1%
`ADP`	10	1.0%
`_`	2	0.2%

Table 2: Top UPOS categories. `PUNCT`, `SYM`, and `INTJ` were not used.

Most of the corpus was annotated collaboratively with continuous revisions to maximize annotation quality given the lack of reliable modern grammars and lexicons for Ashokan Prakrit. Major Rock Edict 7 at Girnar (5 sentences, 49 tokens) was annotated by both authors independently to compute interannotator agreement figures. Agreement scores are reported in table 1b. Agreement on universal POS tagging and head attachment is high. Low labelled attachment score (LAS) reflects the difficulty in analysing the sometimes fragmentary language of the corpus, as is expected in treebanking ancient language corpora (David et al., 2009).

The most common (and thus likely pragmatically unmarked word order, modulo the inscriptional nature of the corpus) in Ashokan Prakrit is subject–object–verb, occuring in half of 24 verbs in the corpus with `nsubj` and `obj` dependents, followed by object–subject–verb with 8 occurrences. SOV is the unmarked word order in most New Indo-Aryan languages as well.

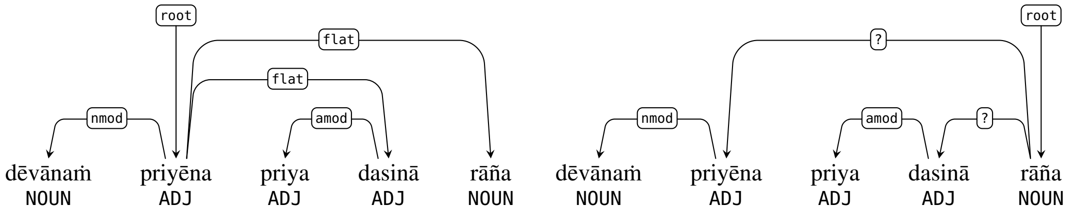
5 Annotation issues

Some of the interesting annotation issues faced include: the POS-tagging and dependency parsing of regnal names in Ashokan Prakrit and cross-lingually (with further discussion on compounds in general), the in-progress transition to split ergativity and its morphological and syntactic analysis within the framework of UD, as well as the relationship between irregular sandhi and the grammaticalization of nouns into adpositions.

A recurring point in the analysis of these issues is that Ashokan Prakrit is transitional between Sanskrit and New Indo-Aryan, still in the process of undergoing many drastic syntactic (from non-configurational to configurational) and morphological (from synthetic to analytic) changes. Given the small size of the corpus and inability to elicit information from native speakers, we faced difficulties annotating features based on a synchronic analysis without looking towards better, and often conflicting, data from Sanskrit or NIA languages.

5.1 Regnal names

A puzzling issue in annotation was Ashoka's regnal names, such as:



(a) `flat` **(b)** any headed dependency

Figure 3: Potential dependency parses (headless and headed) of Ashoka's regnal names.

(2)

Dēvānaṁ-	priyēna	Priya-	dasinā	rāña
god:GEN.M.PL	beloved:INS.M.SG	kindly	looking:INS.M.SG	king:INS.M.SG

'King Beloved-of-the-Gods Looking-Kindly' (Girnar 1:1)

Ashokan Prakrit, like Sanskrit, often constructs chains of nominals and adjectives headed by the last member and with all the members agreeing in case and number with it—here, the instrumental singular. Tokenization, POS-tagging of morphemes in compounds, and dependency relations in regnal names all came up as issues in UD annotation. The decisions in this section were arrived at after much discussion with the UD community.[4]

5.1.1 POS annotation of morphemes in compounds

The first issue was how to POS tag the morphemes in such compounds. In Ashokan Prakrit, like in Sanskrit, "the division-line between substantive and adjective ... [is] wavering" (Whitney, 1889) so any of these titles could be thought of as nominals ('one who is beloved of the Gods') or adjectives ('beloved by the Gods'). Furthermore, syntactic context can blur the distinction; an adjective like *dasin* 'looking' can be nominalized into 'looker', and a noun in a compound may behave attributively.

Initially, we thought to label all the morphemes in the regnal names as `PROPN` given that they refer to a person like a regular name does. However, these morphemes have internal dependency structure, most obviously the genitive-case modifier in *dēvānaṁ-priyēna*. The `PROPN` label would obscure what is clearly a genitive-case `NOUN`, *dēvānaṁ* 'of the Gods', that does not refer to a specific individual or entity like a name does.

In regards to differentiating between `NOUN` and `ADJ` in Ashokan Prakrit, we settled on the criterion that something with a fixed inherent gender must be `NOUN`, and anything with fluid gender assignment is `ADJ`. This makes the POS tag a lexical feature rather than one that is contextually assigned by syntactic properties, which would render it redundant. UD precedent in other languages, e.g. Latin, favours the annotation of dependency structure in proper nouns and the regular POS tagging of nominalized components in such names.[5]

5.1.2 Dependency structure of nominalized titles

There is substantial disagreement among UD corpora on the dependency annotation of regnal names, epithets, and other appellative titles. The current UD guidelines prefer the `flat` relation for "exocentric (headless) semi-fixed MWEs [multi-word expressions] like names and dates". The head is arbitrarily assigned to be the first nominal in the multi-word expression. This is unacceptable for titles in Ashokan Prakrit, since want to treat this the same way as adjective–noun NPs, with the head always being the last word. Schneider and Zeldes (2021) recently attempted to resolve this issue for a wide range of nominal constructions in English (including *Mr.* and *Secretary of State*, which are similar to Ashokan Prakrit titles), and we build upon that analysis here.

Since we have established that in Ashokan Prakrit such constructions are not headless, we have to decide which headed dependency relation should be used instead. We considered `appos`, `compound`, and `nmod:desc`, and `amod` if we chose to analyse the appellatives as adjectival rather than nominal. The difference between a headed and headless dependency analysis of the regnal titles is shown in figure 3.

[4] Documented in a GitHub issue: `https://github.com/UniversalDependencies/docs/issues/802`.

[5] `https://github.com/UniversalDependencies/docs/issues/777`

The issues, resolved once we came to nmod after settling our POS tagging, in the other relations are:

- appos: Generally, an appositive is a full NP that can be paraphrased with an equational copula in a relative clause, e.g. *Bob, my friend* implies *Bob, who is my friend.* But in Ashokan Prakrit, given the blurring between nouns and adjectives, it is clear that each title NP is directly modifying the NP *rāña* 'king' rather than paraphrasing an appositional relationship.
- compound: Like flat, this indicates a multiword expression forming a single NP rather than relationships between full NPs. Each regnal name is, however, a whole NP that could stand alone.
- amod: Our reasoning against the other two relies on analysing each title as an NP. The fact that titles can be dropped, and that *rāña* 'king' can be dropped while retaining grammaticality, supports the assumption that each title is indeed an NP since any one could be the head if phrase-final. Thus, an adjectival relation like amod is not preferred.

Realising that the head of each NP title is lexically a nominalized ADJ, we settled on nmod:desc as the best dependency relation. Further evidence comes from variation in the components of the titles in different editions of the edicts, e.g. (3) and (4). Given that (4) drops 'king' entirely and can have the titles stand alone without another NP head, we are certain that each title is an NP.

(3) Dēvana- priasa raño
god:GEN.M.PL beloved:GEN.M.SG king:GEN.M.SG
'King Beloved-of-the-Gods' (Shahbazgarhi 1:1)

(4) Dēvānaṁ- piyēna Piya- das[i]nā
god:GEN.M.PL beloved:INS.M.SG kindly looking:INS.M.SG
'Beloved-of-the-Gods Looking-Kindly' (Kalsi 1:1)

Now backed with our crosslinguistic evidence, we agree with Schneider and Zeldes (2021) that nmod or a subtyped label of it is the best descriptor for nominal epithets. We specifically picked the subtyped label so that we can query instances of the construction for future analysis.

5.2 Predicated *-ta* construction

The *-ta* construction[6] in Sanskrit forms participles from verbal roots. These participles are morphologically deverbal adjectives, taking gender (without having intrinsic fixed gender like nouns), case, and number marking without marking person (unlike finite verb forms).

(5) rājñā **hataḣ** cauraḣ
king:INS.M.SG kill:PPP.NOM.M.SG thief:NOM.M.SG
'a thief killed by a king' (lit. 'a king-killed thief') (Sanskrit)

In Sanskrit, especially in post-Vedic texts, it can also be interpreted with (past) perfect meaning. *-ta* forms agree in case/gender/number with the object, unlike the finite verbs of this stage of Indo-Aryan.

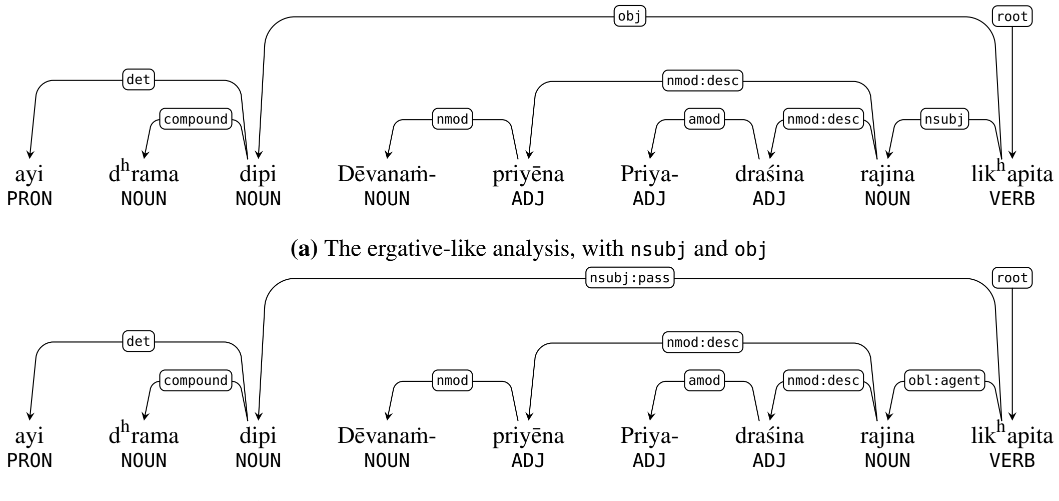
(6) mayā lipī **lik^{h}itā**
1SG.INS text:NOM.F.SG write:PPP.NOM.F.SG
'the text was written by me' (passive)
'I wrote the text' (ergative) (Sanskrit)

This use is extremely common in Ashokan Prakrit and is the point of contention discussed here. According to one view, *-ta* formed resultative[7] adjectives in early OIA, gradually shifting towards main predicate function in first intransitive and later transitive verbs (the agent receiving case marking) by late OIA (Reinöhl, 2018; Condoravdi and Deo, 2014; Peterson, 1998).

This construction is ancestral to the tense/aspect-based split ergativity observed in many later NIA languages. In such languages, the Sanskrit participle has developed into a perfect verb that agrees with the object, while other inflected forms in the verb paradigm agree with the subject. Since Ashokan Prakrit

[6] Philologically known as the *past passive participle*.

[7] As opposed to stative adjectives, resultatives imply that a prior event occurred to cause the current state conveyed by the adjective. Compare English *is hidden* with *has been hidden* (Condoravdi and Deo, 2014).



(a) The ergative-like analysis, with `nsubj` and `obj`

(b) The passive analysis, with `obl:agent` and `nsubj:pass`

Figure 4: Two possible analyses of the predicated *-ta* construction in the sentence '*king Beloved-of-the-Gods Looking-Kindly has caused this rescript on morality to be written*' (Mansehra 1:1). The above was ultimately chosen.

was still undergoing this transition to split ergativity, we could analyze this construction either way: as a resultative predicate adjective or a perfect-aspect verb.

In Ashokan Prakrit, with the loss of the inherited active aorist as a productive category, *-ta* forms have become the unmarked strategy to express the past perfect (Bubeník, 1998). We believe, with some certainty, that this construction is *not passive* at least as late as Ashokan Prakrit (if it ever was). Evidence Casaretto et al. (2020) provide against a passive analysis in Sanskrit also applies here. A key argument is that *-ta* occurs with both transitive and intransitive verbs, and in the case of the latter, does not form an "impersonal passive" as would be expected of a passivized intransitive.

As such, we adopt an ergative-like analysis of the *-ta* construction in Ashokan Prakrit, agreeing with Peterson (1998)'s view of the corresponding construction in Pali (another early MIA lect) as being a periphrastic perfect. Indeed, as exhibited in the example in figure 4a which is glossed in (7), the *-ta* form agrees in number and gender with the object, while the agent receives instrumental marking. The object *d^{h}rama-dipi* is still in the nominative case.

(7) ayi d^{h}rama- dipi Dēvanaṁ- priyēna Priya- draśina
DEM3:F.SG morality rescript:NOM.F.SG god:NOM.M.PL beloved:INS.M.SG kindly looking:INS.M.SG
rajina lik^{h}apita
king:INS.M.SG write:CAUS.PPP.NOM.F.SG
'King Beloved-of-the-Gods Looking-Kindly has caused this rescript on morality to be written' (Mansehra 1:1)

With respect to UD annotation, our ergative-like analysis translates to the agent *rajina* receiving the `DEPREL` `nsubj` and the object *dipi* `obj` (instead of `obl:agent` and `nsubj:pass` of the passive analysis in figure 4b).

5.2.1 Differential agent marking

Cross-dialectally as well as dialect-internally, Ashokan Prakrit varies with respect to how the agent phrase is marked in *-ta* constructions. Agents may receive either **instrumental** or (with lesser frequency) **genitive** case marking, though the basis for this alternation is not wholly clear.

(8) sē **mamayā** bahu kayānē kaṭē
now 1SG.**INS** many good_deed:NOM.N.SG do:PPP.NOM.N.SG
'Now, I did many good deeds.' (Kalsi 5:4)

(9) Dēvānaṁ- piya**śa** Piya- daś**inē** lāj**inē** Kaligyā
god:NOM.M.PL beloved:**GEN**.M.SG kindly looking:**GEN**.M.SG king:**GEN**.M.SG Kalinga:NOM.M.PL
vijitā
conquer:PPP.NOM.M.PL

'... king Beloved-of-the-Gods Looking-Kindly conquered the Kalingas.' (Kalsi 13:1)

Anderson (1986)'s analysis suggests discourse-pragmatic factors may be at play; the genitive agent conveys old (i.e. contextually given and/or definite) information while the instrumental agent conveys new information. On this basis, he also claims these represent two *separate* constructions, a passive and an ergative respectively, though this proposal has some flaws (see (Bubeník, 1998) for criticisms).

We tentatively follow Dahl and Stroński (2016) in analyzing the situation as one of **differential agent marking (DAM)** (Arkadiev, 2017), whereby two agent-marking cases are distributed along (potentially irrecoverable) semantic/pragmatic lines. Thus we stuck with standard morphological analysis of the case features in these agents, i.e. `Case=Gen/Ins` rather than explicitly proposing `Case=Erg` as an Ashokan Prakrit feature.

DAM seems to affect both the agents of the ergative-like *-ta* construction as well as the oblique agents of finite passives in Ashokan Prakrit. Of the source constructions in Vedic, Bubeník (1998) explains there is a broad tendency for 'active' verbs to favor instrumental agents, and 'ingestive' verbs (perception, consumption, etc.) to favor the genitive, but the instrumental becomes default in later stages of Old Indo-Aryan. Further annotation of the Ashokan Prakrit corpus will allow us to probe into these hypotheses with statistical tools.

Additionally, the influence of Ashoka's administrative language, an eastern dialect from which other dialectal edicts were likely translated (Oberlies, 2003), should not be neglected. If the choice between instrumental and genitive marking is at least partially a function of dialect, direct translation from Ashoka's variety could leave relic forms[8] (otherwise inconsistent with the internal distribution of cases) in other edicts.

5.3 Sandhi

Sanskrit texts (which in written form all post-date the Ashokan edicts) generally orthographically indicate *sandhi*, a kind of phonological assimilation at morpheme boundaries (Allen, 1962). Some examples from Sanskrit are given in (10).

(10) a. gacc^{h}at**i** arjunaḣ → gacc^{h}at**y**arjunaḣ (Sanskrit)

b. s**aḣ** **a**ham → s**ō'**ham

c. brahm**a** **a**smi → brahm**ā**smi

Middle Indo-Aryan has more haphazard orthographic indication of sandhi rules (Dočkalová, 2009), even though these assimilations likely persisted in speech. For example, Pali shows sandhi in compounds (especially those inherited directly from Old Indo-Aryan and then subject to normal sound changes), some function words (emphatic *ēva*, preverbs, etc.), pronouns, and sometimes in nominal arguments to verbs, noun–noun relations, and vocatives (Childers, 1879). That is, Pali optionally indicates sandhi only between syntactically related words (Oberlies, 2001, p. 116).

We observed similar occurrences in the Ashokan Prakrit corpus. We think certain rare cases of sandhi in Ashokan Prakrit may be examples of grammaticalization (the development of a postposition with case-like properties) and lexicalization (compounds that are no longer as transparent). These pose issues for UD annotation.

5.3.1 Grammaticalization of *at^{h}āya ~ aṭ^{h}āya*

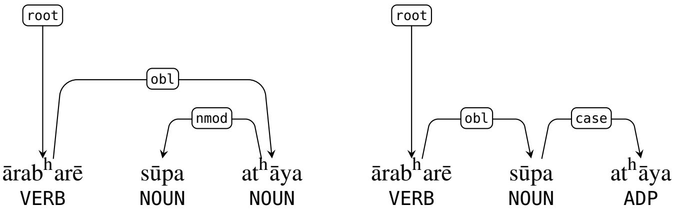
One case where sandhi may gives us clues about morphological change is occurrences of *at^{h}āya* 'for the purpose [of]', the dative of *at^{h}a* 'purpose' (< Sanskrit *árt^{h}a*). In the prototypical example below, sandhi with the preceding nominal stem causes vowel lengthening.

(11)

tī	ēva	prāṇā	ārab^{h}arē	sūp-	**āt^{h}āya**
three	EMPH	animal:NOM.N.PL	kill:PASS.PRS.IND.3PL	curry	purpose:DAT.M.SG

'Only three animals are being killed for the purpose of curry.' (Girnar 1:7)

[8] One such example of dialectal interference is NOM.M.SG *-ē*, a non-western isogloss, attested in place of the expected -ō in Girnar (a western dialect)



(a) normal NP **(b)** grammaticalized `case`

Figure 5: Two potential analyses of the *at^{h}āya* construction in Girnar 1:7.

While the Ashokan Prakrit construction we are dealing with is a compound,[9] not a genitive noun modifier, Reinöhl (2016) describes a potentially related phenomenon based on Classical Sanskrit and Pali corpora: the **post-Vedic genitive shift**, wherein many adverbs and adjectives were analysed as taking the genitive and periphrastically replacing case relations, e.g. *-asya art^{h}āya* 'for the purpose of ...'. Sanskrit generally uses the dative case by itself to indicate PURPOSE, so this compounded construction in Ashokan Prakrit may be an intermediate phase in the genitive shift.

For UD, this is a tricky situation. We were stuck between describing *at^{h}āya* as a `case` complement to *sūpa*, or as instead the head of an NP, both shown in figure 5. Given similar constructions with ending-less nouns in compounds, *at^{h}āya* would usually be analysed as the head here, but if it has been grammaticalized then `case` would be a better `DEPREL` as is used for case markers and clitics in New Indo-Aryan UD, and UD prefers content heads. Girnar 4:10 also has *ētāya at^{h}āya* 'for this purpose' which lacks sandhi or stem-compounding, but this may be exceptional since *ētad* can take the `det` `DEPREL` as a modifier to nouns and so does not behave like a true nominal. Pending better evidence supporting either analysis, we settled on the latter.

An interesting data point is that a similar construction is the etymological source of the dative case in the Insular Indo-Aryan languages, Dhivehi and Sinhala.

(12) mamma e=ge-**aṣ̊** diya
mother DEM3=house-**DAT** go.PST.ALTER
'Mother went to that house.' (adapted from Lum, (2020): 118) (Dhivehi)

(13) ammā ē gedərə-**ṭə** giyā
mother DEM4 house.SG-**DAT** go.PST
'Mother went to that house.' (Sinhala)

The Sinhala *-(ə)ṭə* and Dhivehi *-aṣ̊* datives are both reflexes of Sanskrit *árt^{h}āya* (or, possibly, the accusative case *árt^{h}aṁ*) (Fritz, 2002) and have expanded their semantic domains to include other roles such as GOAL. Ashokan Prakrit's compounding of *at^{h}āya* may represent an early stage towards a similar grammaticalization, though its precise synchronic status is unclear. Future UD annotation of MIA corpora will allow us to better track such phenomena from a comparative perspective.[10]

5.3.2 Other cases

Another unexpected sandhi was observed in Girnar 2:2, *manusōpagāni ca pasōpagāni ca* 'beneficial to man and beneficial to animal'. The form *pasōpagāni* is underlying *pasu* '(domestic) animal' + *upagāni* 'benefits', wherein the sandhi of *u* + *u* gives *ō* rather than expected *ū* (as in Sanskrit) or *u* (as in Pali). This sandhi is found in every other edition of the edict; Jaugada even has *pasu-ōpagāni*. Like the previous

[9] A similar construction involving "compounded" *art^{h}āya* also occurs in certain Sanskrit texts, cf. *harṣaṇārt^{h}āya* |*harṣaṇa* + *art^{h}āya*| 'for the purpose of protection'(Fritz, 2002).

[10] It is worth noting that an inscriptionally-attested Middle Indo-Aryan ancestor of Sinhala, roughly contemporaneous with the Ashokan edicts, formed a periphrastic dative of purpose with *aṭaya* (cf. *śagaśa aṭaya* 'for the benefit of the sangha') (Premaratne, 1969; Paranavithana, 1970). Here, as is also observed with Pali's *att^{h}āya* construction (Reinöhl, 2016; Fahs, 1989), the nominal *śaga* 'sangha' takes genitive case marking. In contrast, Ashokan Prakrit employs either a dative dependent (e.g. *etāya*) or the stem-compounding strategy described above. It cannot be ruled out, however, that the modern Sinhala and Dhivehi datives originate in a similar compound-like use of *árt^{h}āya* (Fritz, 2002).

example, we could claim that *upagāni* is undergoing grammaticalized to a benefactive postposition here, but we feel it is too speculative to claim that, and instead believe it to be phonological analogy with *manusōpagāni*. We analysed it as a noun compound with `DEPREL` `nmod`.

6 Future work

The main task ahead of us involves completing annotation, which will require gathering and critical editing of Ashokan texts discovered in the past century that are yet to be digitally compiled. What has been annotated already will be included within the next annual UD corpus release.

After a good selection of annotated inscriptions from several dialects is available, we will make use of computational methods to analyze the corpus. Automatic word-level alignment between dialectal variants of the same edict will enable us to compare dependency structure, case marking, sound change outcomes, along with other dialectal features. On the technical side, we would also like to see if training data from Sanskrit with finetuning on the smaller Ashokan corpus could be used to automatically perform UD annotation of texts in other Middle Indo-Aryan languages.

More broadly, we would like to continue UD annotation of texts in earlier Indo-Aryan languages in order to have data to better address historical linguistic questions. Given the value already demonstrated by corpus data for Indo-Aryan historical linguistics (Stroński and Verbeke, 2020), open-access corpora annotated using Universal Dependencies, with fine-grained analyses of morphology and syntax beyond individual glossed examples, will surely help put some of the controversial issues in the field to rest. Comparisons of Ashokan Prakrit with other stages of Indo-Aryan will help us study language change, e.g. the development of configurationality in Middle Indo-Aryan (Reinöhl, 2016). Dialectal variation (and possible substrate influence) in Ashokan Prakrit should also be studied in comparison with regional NIA data. Other recent work in computational approaches to this area (Cathcart and Rama, 2020; Cathcart, 2020; Arora et al., 2021; Arora and Farris, 2021) encouraged us to pursue the study of South Asian historical linguistics from a similar angle.

Some texts we hope to treebank in the future include: the Pāli Canon, plays in the various later Dramatic Prakrits (e.g. *Gāhā Sattasaī*), the *Lōmāfānu* documents (Old Dhivehi), the *Bāṇāsurakat^{h}ā* (Old Kashmiri), the *Gurū Grant^h Sāhib* (Sant Bhāṣā, Old Punjabi), the *Caryāpada* (Old Bengali), the *Šāh jō Risālō* (Sindhi), and epics and poetry from the Hindi Belt and Maharashtra. Serious work on typology in South Asia will also require treebanking for Dravidian (which has a long historical attestation), Munda, and other language families of the region.

Acknowledgements

We thank John Peterson for his continuous guidance in our work on Ashokan Prakrit, and Amir Zeldes, Nathan Schneider, and Samopriya Basu for their helpful comments on the manuscript. We also thank the three anonymous UDW reviewers for their constructive feedback.

References

W. Sidney Allen. 1962. *Sandhi: The theoretical, phonetic, and historical bases of word-junction in Sanskrit*. De Gruyter Mouton, Boston.

Paul Kent Anderson. 1986. Die ta-partizipialkonstruktion bei aśoka: Passiv oder ergativ? *Zeitschrift für vergleichende Sprachforschung,*, 99(1):75–94.

Peter Arkadiev. 2017. Multiple ergatives: From allomorphy to differential agent marking. *Studies in Language*, 41(3):717–780.

Aryaman Arora and Adam Farris. 2021. *Jambu*. Georgetown University, Washington.

Aryaman Arora, Adam Farris, Gopalakrishnan R, and Samopriya Basu. 2021. Bhāṣācitra: Visualising the dialect geography of South Asia. In *Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021*, pages 51–57, Online, August. Association for Computational Linguistics.

Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for Hindi/Urdu. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pages 186–189, Suntec, Singapore, August. Association for Computational Linguistics.

Jules Bloch. 1950. *Les inscriptions d'Asoka*. Société Parisienne d'Édition, Paris.

Jens Braarvig, Asgeir Nesøen, and University of Oslo. 2014. The Ashoka Library.

Vít Bubeník. 1998. *A Historical Syntax of Late Middle Indo-Aryan (Apabhraṃśa)*. John Benjamins, Amsterdam & Philadelphia.

Antje Casaretto, Gerrit J. Dimmendaal, Birgit Hellwig, Uta Reinöhl, and Gertrud Schneider-Blum. 2020. Roots of ergativity in Africa (and beyond). *Studies in African Linguistics*, 49(1):111–140.

Chundra Cathcart and Taraka Rama. 2020. Disentangling dialects: a neural approach to Indo-Aryan historical phonology and subgrouping. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 620–630, Online, November. Association for Computational Linguistics.

Chundra A. Cathcart. 2020. A probabilistic assessment of the Indo-Aryan Inner–Outer hypothesis. *Journal of Historical Linguistics*, 10(1):42–86.

R. C. Childers. 1879. On sandhi in Pali. *The Journal of the Royal Asiatic Society of Great Britain and Ireland*, 11(1).

Cleo Condoravdi and Ashwini Deo. 2014. Aspect shifts in Indo-Aryan and trajectories of semantic change. In Chiara Gianollo, Agnes Jäger, and Doris Penka, editors, *Language Change at the Syntax-Semantics Interface*, pages 261–292. De Gruyter Mouton.

Eystein Dahl and Krzysztof Stroński. 2016. Ergativity in Indo-Aryan and beyond. In Eystein Dahl and Krzysztof Stroński, editors, *Indo-Aryan Ergativity in Typological and Diachronic Perspective*, pages 1–37. John Benjamins, Amsterdam & Philadelphia.

Bamman David, Francesco Mambrini, and Gregory Crane. 2009. An ownership model of annotation: The Ancient Greek dependency treebank. In *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories (TLT 8)*, pages 5–15, Groningen.

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

Lenka Dočkalová. 2009. Development of sandhi phenomena in sanskrit and aśōkan prakrit and pāli. *Linguistica Brunensia*, 57(1-2):45–59.

Puneet Dwivedi and Daniel Zeman. 2017. Universal Dependencies for Sanskrit: A pilot study. Preprint.

Achim Fahs. 1989. *Grammatik des Pali*. Enzyklopädie, Leipzig.

Sonja Fritz. 2002. *The Dhivehi Language. A Descriptive and Historical Grammar of Maldivian and Its Dialects*. Ergon-Verlag, Würzburg.

Oliver Hellwig, Salvatore Scarlata, Elia Ackermann, and Paul Widmer. 2020. The treebank of vedic Sanskrit. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5137–5146, Marseille, France, May. European Language Resources Association.

Gérard Huet. 2005. A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger. *Journal of Functional Programming*, 15(4):573–614.

E. Hultzsch. 1925. *Corpus Inscriptionum Indicarum*. Clarendon Press, Oxford.

David Kopf. 1969. *British Orientalism and the Bengal Renaissance: The Dynamics of Indian Modernization 1773–1835*. University of California Press.

Amba Kulkarni, Pavankumar Satuluri, Sanjeev Panchal, Malay Maity, and Amruta Malvade. 2020. Dependency relations for Sanskrit parsing and treebank. In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, pages 135–150, Düsseldorf, Germany, October. Association for Computational Linguistics.

Jonathon Lum. 2020. An egophoric analysis of Dhivehi verbal morphology. In Henrik Bergqvist and Seppo Kittilä, editors, *Evidentiality, egophoricity, and engagement*, pages 95–139. Language Science Press, Berlin.

Madhukar Anant Mehendale. 1948. *Historical Grammar of Inscriptional Prakrits*. Deccan College, Postgraduate and Research Institute, Poona.

M. Monier-Williams. 1899. *A Sanskrit–English dictionary: Etymologically and philologically arranged with special reference to cognate Indo-European languages*. The Clarendon Press, Oxford.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Thomas Oberlies. 2001. *Pāli: A Grammar of the Language of the Theravāda Tipiṭaka*. De Gruyter, Berlin & New York.

Thomas Oberlies. 2003. Aśokan Prakrit and Pāli. In George Cardona and Dhanesh Jain, editors, *The Indo-Aryan Languages*, pages 161–203. Routledge, London & New York.

Senarat Paranavithana. 1970. *Inscriptions of Ceylon: Containing cave inscriptions from 3rd century B.C. to 1st century A.C. and other inscriptions in the early Brāhmī script*. Department of Archaeology, Sri Lanka., Colombo.

John Peterson. 1998. *Grammatical Relations in Pali and the Emergence of Ergativity in Indo-Aryan*. Lincom, Munich.

Asoka Premaratne. 1969. *The Verb in Early Sinhalese*. Ph.D. thesis, School of Oriental and African Studies, University of London.

Vinit Ravishankar. 2017. A Universal Dependencies treebank for Marathi. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 190–200, Prague, Czech Republic.

Uta Reinöhl. 2016. *Grammaticalization and the Rise of Configurationality in Indo-Aryan*. Oxford University Press.

Uta Reinöhl. 2018. Eystein Dahl and Krzysztof Stroński (eds.). Indo-Aryan ergativity in typological and diachronic perspective (Typological Studies in Language, 112). *Journal of South Asian Languages and Linguistics*, 5(1).

Salvatore Scarlata, Oliver Hellwig, Elia Ackermann, Erica Biagetti, and Paul Widmer. 2020. Annotation guidelines for the Vedic Treebank v. 1. Preprint.

Nathan Schneider and Amir Zeldes. 2021. Mischievous nominal constructions in Universal Dependencies.

David A Scott. 1985. Ashokan missionary expansion of Buddhism among the Greeks (in NW India, Bactria and the Levant). *Religion*, 15(2):131–141.

Amulyachandra Sen. 1956. *Asoka's Edicts*. Pooran Press, Calcutta.

D. C. Sircar. 1966. *Indian epigraphical glossary*. Motilal Banarsidass, Delhi.

Monica L. Smith, Thomas W. Gillespie, Scott Barron, and Kanika Kalra. 2016. Finding history: the locational geography of Ashokan inscriptions in the Indian subcontinent. *Antiquity*, 90(350):376–392.

Krzysztof Stroński and Saartje Verbeke. 2020. Shaping modern indo-aryan isoglosses. *Poznan Studies in Contemporary Linguistics*, 56(3):529–552.

Juhi Tandon, Himani Chaudhry, Riyaz Ahmad Bhat, and Dipti Sharma. 2016. Conversion from paninian karakas to Universal Dependencies for Hindi dependency treebank. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 141–150, Berlin, Germany, August. Association for Computational Linguistics.

Francis M. Tyers, Mariya Sheyanova, and Jonathan North Washington. 2017. UD annotatrix: An annotation tool for Universal Dependencies. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 10–17, Prague, Czech Republic.

Siddheshwar Varma. 1947. "Historical Grammar of Inscriptional Prakrits" by M. A. Mehendale. *Annals of the Bhandarkar Oriental Research Institute*, 28(3/4):320–325.

William Dwight Whitney. 1889. *Sanskrit Grammar Including Both the Classical Language and the Older Dialects of Veda and Brahmana*. Harvard University Press.

Alfred C. Woolner. 1924. *Asoka Text and Glossary*. Oxford University Press.

UD on Software Requirements: Application and Challenges

Naïma Hassert[*]
Université de Montréal
naima.hassert@umontreal.ca

Pierre André Ménard[*]
CRIM
menardpa@crim.ca

Edith Galy
CRIM
galyed@crim.ca

Abstract

Technical documents present distinct challenges when used in natural language processing tasks such as part-of-speech tagging or syntactic parsing. This is mainly due to the nature of their content, which may differ greatly from more studied texts like news articles, encyclopedic extracts or social media entries. This work contributes an English corpus composed of software requirement texts annotated in Universal Dependencies (UD) to study the differences, challenges and issues encountered on these documents when following the UD guidelines. Different structural and linguistic phenomena are studied in the light of their impact on manual and automatic dependency annotation. To better cope with texts of this nature, some modifications and features are proposed in order to enrich the existing UD guidelines to better cover technical texts. The proposed corpus is compared to other existing corpora to show the structural complexity of the texts as well as the challenge it presents to recent processing methods. This contribution is the first software requirement corpus annotated with UD relations.

1 Introduction

Since its first release (Nivre et al., 2016), the Universal Dependencies (UD) treebank project has grown to over 200 repositories across 114 languages as of version 2.8 (Nivre et al., 2020). These treebanks target various types of documents such as news, fiction, grammar examples, spoken transcription, nonfiction, Wikipedia content, legal, religious, fiction, social media, blog, email, poetry, medical, web pages, academic, government and essays. While these types of text are often encountered, this selection leaves out one important subgenre of nonfiction: technical documents. From a natural language processing (NLP) perspective, having an annotated corpus is essential to study, evaluate and potentially train and optimize machine learning algorithms to process a specific type of text.

As a first step to study and evaluate their content, this work focus on the study of technical documents through the exploration of software requirements (SR) specifications. These types of documents often deviate from standard free-flowing text, hindering manual as well as automatic analysis of universal dependencies.

This article presents the contextual background of the study in the next section. Section 3 describes how the raw corpus was constructed, while Section 4 enumerates some phenomena that were observed in the corpus and how they were annotated. Section 5 compares the new corpus with other existing English UD annotated corpora.

2 Problem context

Technical documents, a subclass of nonfiction documents, can take several roles or forms. They can be instruction manuals, equipment maintenance procedures, documentation of schematics or plans, and so on. They might contain images, schemas, and isolated or large sections of texts, depending on their focus. Their goal might be distilled as conveying specific information in a clear, concise and unambiguous

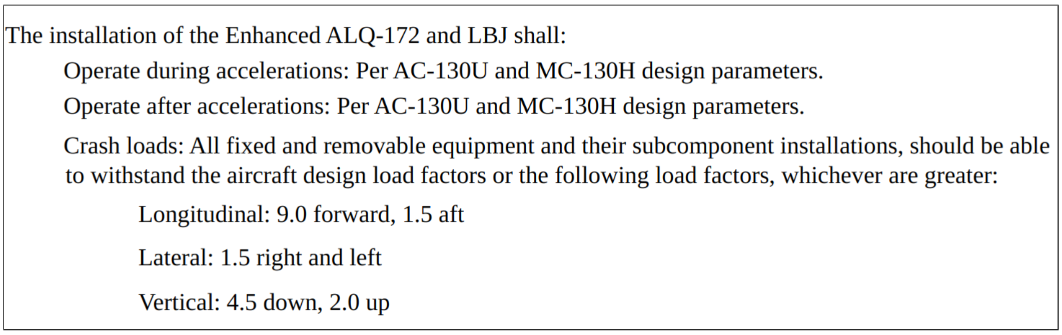
[*] These two authors contributed equally to this work as first authors.

way. Among this subgenre exists a specific type of technical document called software requirements specification (SRS).

Technical documents, and more specifically software requirements (SR) specifications, are written with the goal to inform the reader about a subject by giving information that is clear, measurable and unambiguous. Software requirement specifications are broken into multiple software requirements (SR). These can be analysed by software experts with the goal to unambiguously understand them and develop a software system that fulfills their needed functionalities.

Natural language processing tasks can be applied to almost any step of the software requirement life cycle, like elicitation, analysis, modeling, verification, etc (Zhao et al., 2021). Applied to software requirement specifications, dependency analysis can have multiple applications. One such use in analysis pipeline is to help perform semantic parsing (Roth and Klein, 2015) or semantic frame parsing (Wang, 2016) on SR by linking tokens to their governor, up to the root of the sentence. This step can support semantic parsing in detecting relevant parts of the sentence and attributing them specific roles such as actor, object, condition, action, etc. The result of this analysis can be used to automatically generate test cases (Ahsan et al., 2017) in order to verify software systems and improve them. Improving dependency analysis on technical documents and, more specifically, SR, can enhance the overall performance of those tasks.

Unfortunately, this is easier said than done. Several types of structural and linguistic phenomena hinder the progress of automatic dependency parsing. Manually parsing dependency relations for these texts is a difficult task for any human annotator for two reasons: the text's related technical expressions are not intuitive to understand and there are some limitations to the application of the UD guidelines. While human experts in the domain rarely have issues interpreting SR described in natural language, NLP tools trained on free-flowing texts have more difficulty in interpreting and linking sentence segments to perform dependency parsing.



The installation of the Enhanced ALQ-172 and LBJ shall:

Operate during accelerations: Per AC-130U and MC-130H design parameters.

Operate after accelerations: Per AC-130U and MC-130H design parameters.

Crash loads: All fixed and removable equipment and their subcomponent installations, should be able to withstand the aircraft design load factors or the following load factors, whichever are greater:

Longitudinal: 9.0 forward, 1.5 aft

Lateral: 1.5 right and left

Vertical: 4.5 down, 2.0 up

Figure 1: Source document representation of a sample software requirement for a radio system.

Figure 1 shows an example of such software requirement for a radio system designed for a specific type of airplane. Briefly looking at this example, one can see multiple atypical phenomena when compared to traditional texts: partial sentences, vertical enumerations, acronyms, domain-specific named entities, etc. While this SR does not represent the majority of requirement texts in any given SRS, its composition and presentation format are common in this type of document, especially in systems that manage scientific processes or interact with physical modules. As automatic dependency parsers are usually trained on complete and well-structured sentences, they behave erratically on such texts. They fail to assign correct part-of-speech (POS) tags and to accurately detect the heads and their corresponding dependency relations. This motivates the development of annotated corpora in order to better study the represented phenomena and develop better approaches to correctly analyze them.

3 Corpus Creation Process

One of the goals for creating this SR corpus (hereafter named CTeTex for CRIM's Technical Texts corpus) was to study the performances of automatic dependency parsing on uncommon SR texts that typically cause issues with these tools.

The corpus was composed of source documents from public sources. These were taken from different websites, including one previously released dataset, which is called the PURE software requirements corpus (Ferrari et al., 2017). This last repository was used as a source to extract the majority of SR for the corpus. It contained SRS documents ranging in complexity from student projects for a multiplayer game to telescope grid communication software. The documents were produced or owned by public organizations so their licence permits free use for academic research.

Following the goal of this corpus, the requirements were chosen based on their particular attributes in order to present various features of software requirement documents and, more broadly, technical documentation. Documents from the corpus were scanned manually in order to differentiate SR from non-requirements segments (section's introduction, generic explanations, etc.) and were sampled based on the expected challenge they offered. While a random selection would have been more representative of the given corpus, this selection process gives a better view of the problematic issues presented by these types of documents.

The Inception platform (Klie et al., 2018) was used to perform dependency annotation. As it performs default tokenization on texts, tokens were corrected manually. Basic UD dependency relations were added to the tokens without the enhanced version. The resulting corpus[1] contains 196 SR coming from 24 source documents with 9,273 tokens distributed in 276 sentences. An annotator syntactically analyzed each document to produce token segmentation, part-of-speech tags and labelled dependency relations. The annotator was a student mastering in linguistics with prior knowledge of dependency grammar and was trained with a senior linguist on an alternate proprietary technical corpus to ensure the quality of the annotations. The initial Kappa inter-annotator agreement on the first segment of the training corpus was 0,69, but grew to near-perfect match on non-problematic cases through iterative consulting with the senior linguist. This ensured that the UD guidelines were understood and homogeneously applied by the annotator and the senior linguist.

For the CTeText corpus, the annotator consulted with a software engineer in order to clarify technical terms and expressions that were specific to the software or engineering domain, a type of system or a single system. This helped clarify some ambiguity on both part-of-speech tags and dependency relations. The annotations were then revised by a group of three experts (including the annotator) to discuss unresolved cases in order to solve them. The annotation process took approximately 180 hours, considering only the main annotator's time.

4 Applying UD Guidelines on Software Requirements

This section describes various structural and linguistic phenomena that are often found in SR texts, the issues in applying the guidelines and how they were annotated in the proposed corpus. Table 1 shows the quantity of examples found in the corpus for each of the subsections. Note that some text segments, like a scientific notation using abbreviations and specialized vocabulary, can be counted in multiple categories.

Phenomena	Occurrences	Number of tokens
Scientific and mathematical notations	22	68
Abbreviations and acronyms	579	579
Lists and enumerations	41	2 736
Specialized vocabulary	503	1 180

Table 1: Estimation of occurrences and their total number of tokens for each phenomenon in CTeTex.

For each issue detected during the annotation phase, an in-depth analysis was done in order to find the most adequate proposition. Each of the three experts revisited the relevant UD definitions of all the possible alternatives to solve the issue. They then looked at the other English corpora (referred in Section 5) with the Grew-match[2] online search tool to check if there was similar cases that could

[1] The corpus will be published on the Universal Dependencies repository under CTeTex name.

[2] `http://match.grew.fr`

shed some light on the targeted challenge and if they could apply to the context of CTeTEx corpus. Specialized online discussions on universal dependencies were also consulted when topics aligned with one or multiple possible solutions. Individual findings were then discussed together in order to review the possible options and evaluate which one was the best to express the syntactic aspect in the specialized context of technical texts.

4.1 Scientific and Mathematical Notations

Mathematical formulas and scientific expressions are often found in SR texts in order to inform the reader on the valid response the described system should provide. However, the UD documentation does not address which dependency relations are appropriate for those specific types of construction. One of the only referenced cases related to those expressions is the specification that mathematical operators should be designated as a *SYM* POS tag. The following sections provide details about the annotation choices made for the CTeTex corpus as a well as a proposition to enhance the UD guidelines.

4.1.1 Formulas

Formulas use variables, coefficients and operators to clearly communicate a scientific or mathematical process that should be implemented by a system, as shown in the underlying sentence in Figure 2. In an attempt to determine which dependency relations were adequate in these cases, the verbalization of those mathematical formulas and expressions was first considered. What could one say if one wanted to express the formula in words? It was then proposed that the symbol "=" could be considered as the equivalent of the verb "equals" or "is", as the head of the expression, the symbol "+" as the coordinating conjunction "plus" or "and", etc.

However, the verbalization of those expressions is far from straightforward. For example, the expression $O(n^2)$. If the annotator does not have a background in mathematics or computer science, it could be difficult to come up with a valid verbalization that would translate to *"a big O of n squared"*[3].

The solution proposed is thus to acknowledge that mathematics is a formal language and, by nature, does not obey the syntactic rules of natural languages like English. These are in fact two different languages. When the two of them are found in the same text, it is a case of code-switching. Trying to analyze both of them in the same way could result in misleading annotations.

Fortunately, UD already has a way to deal with foreign languages: the relation *flat:foreign*. It is suggested that the head of the mathematical expression should be what is considered to be its first token, for simplicity reasons. In accordance with UD guidelines, this head should be the parent of all the following tokens constituting the mathematical expression. An example of the resulting tree can be seen in Figure 2.

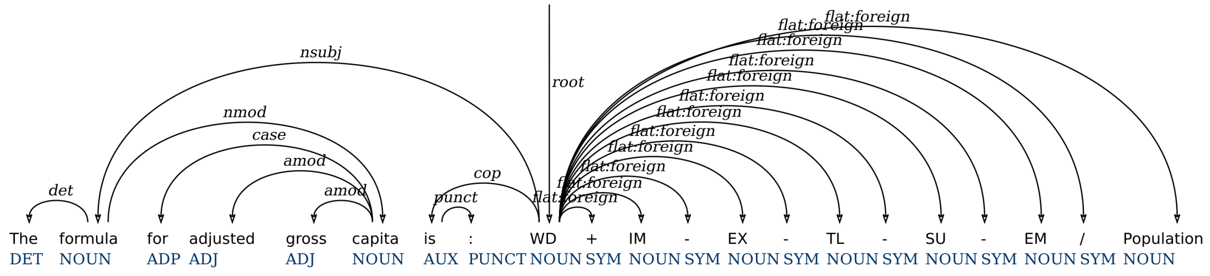

nsubj
nmod
case
amod
amod
det
cop
punct
root
flat:foreign
The formula for adjusted gross capita is : WD + IM - EX - TL - SU - EM / Population
DET NOUN ADP ADJ ADJ NOUN AUX PUNCT NOUN SYM NOUN SYM NOUN SYM NOUN SYM NOUN SYM NOUN SYM NOUN

Figure 2: Annotated mathematical expression.

This solution has several advantages; it does not require the annotator to verbalize expressions that are not intended to be expressed in words, it is constant and easily applicable to any specific case, it uses a dependency relation that already exists in the UD documentation, and it differentiates them from text written in natural language in the corpus.

However, the presence of mathematical operators does not necessarily indicate the presence of a mathematical formula or expression. Sometimes, the equal sign is found alone and is used to indicate the

[3] See https://en.wikipedia.org/wiki/Big_O_notation

meaning of a word or a group of words. In that case, the verbalization approach is more appropriate. As it only implicates a single operator in this case, the proposed solution is to treat the equals sign as a verb. The subject is the word that precedes it and the object is the word that follows it. This is illustrated in Figure 3. Note the error in the sentence : ’groundwater’ should be one word (see ’goeswith’ between ’ground’ and ’water’). That is why ’ground’ is coordinated with ’water’ differently than ’surface’.

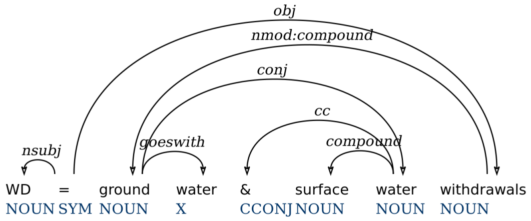


Figure 3: Example of a standalone equal operator.

4.1.2 Variables

Mathematical formulas suggest that variables and placeholders for undefined values will be present in SR texts. For example, in *“less than t minutes”* or *“for n nodes”*. Regarding the part-of-speech tag that should be attributed to variables, the UD documentation does not provide an easy solution. It mentions: “The universal POS tags should capture regular, prevailing syntactic behavior, as well as morphological characteristics when available, and should not reflect sentence-specific exceptional behavior.”

But what is the “prevailing syntactic behavior” of a letter? In the online Merriam-Webster dictionary, letters are considered as nouns when they designate the letter in the alphabet or when used as an abbreviation for a noun that begins with that letter. Nothing is mentioned for when it is a name of a variable. Two choices were thus considered: the UPOS *X*, since no official answer yet exists , or *NUM*, since it is clear that it is what the variable expresses. UD guidelines specify that the UPOS *X* should be used with sparingly, and only when there is no other possibilities. The second solution was thus the one adopted. The dependencies are then defined as the sentence dictates, as if a number was replacing the variable. While there is no case in the proposed corpus, a variable with a different type of implied value (categorical, boolean, string, etc) would be tagged as if a specific value was used, likely with NOUN (i.e. *The system will send the s string”*).

Is it important to note that this proposition excludes named system variables like in ”display the content of the XYZ field” in which the name of the variable is a recognized concept of a system and is named to differentiate it from other similar concepts, thus behaving like a noun and were tagged with PROPN. This is different from an unnamed variable like ”display the top n results” which behaves syntactically like its underlying numerical value.

4.2 Abbreviations and Acronyms

Software requirements usually contain high number of acronyms, mainly from computer science but also from the described system’s application domain. Specifically, acronyms of proper and common nouns, abbreviations of short expressions and Latin abbreviations have been found in the corpus. But information on acronyms and abbreviations is very scarce in UD documentation. This section presents an analysis for these types of constructions. For easier reference, the relevant expressions in the section are provided with the *Abbr=Yes* feature in the treebank.

4.2.1 Acronyms

As mentioned in section 2, software requirements are written with the goal of informing the reader about the software in clear and precise language. That is why this type of document refers to various components of software that often have long, specific and repetitive names. It is then normal that in order to reduce the length of the text and to facilitate its reading, those names are shortened in the form of acronyms.

These types of acronyms are a challenge for an annotator for two reasons. First, the UD documentation does not give clear guidelines for the appropriate POS tag to give to acronyms. It only mentions one case: "Acronyms of proper nouns, such as UN and NATO, should be tagged PROPN." Nothing is said about acronyms of common nouns or acronyms of nominals of which the head is a common noun, like "ID" ("Identifier") or "ETA" ("Estimated Time of Arrival"). It was then decided that acronyms of common nouns should be tagged *NOUN*.

There is a second challenge for the annotator: to find, in the document itself or elsewhere if it is not specified in it, the long form of the acronym. The alternative use of short and long forms of acronyms, without explicit association, is common in organization's internal documents (Ménard and Ratté, 2010). It is crucial in order to understand the general meaning of the requirement, but also for deciding if the appropriate POS tag is *PROPN* or *NOUN*.

It should be noted that the difference between proper and common nouns can be very subtle, and it was not the objective of this paper to address this problem. However, it was decided that within the corpus, proper nouns are nouns that designate a specific entity that can be distinguished in a group of similar entities. The following requirement can be taken as an example.

> The TCS will provide the hardware and software necessary to allow the operator to conduct the following major functions 1) mission planning, 2) mission control and monitoring, 3) payload product management, 4) targeting, and 5) C4I system interface.

"TCS" here means "Tactical Control System". It designates a specific system that can be distinguished in a group of similar systems and that the entire technical document is meant to describe. It is then annotated as a proper noun. However, "C4I" is a type of system, not a specific system: it is then a common noun.

4.2.2 Abbreviations of Short Expressions

When short expressions were abbreviated, like "TBD" ("To Be Determined") or "TBC" ("To Be Confirmed"), it was decided to give them the POS tag of the head of the expression (*VERB* in the mentioned examples). This is because they could easily have been written in their long form and the normal syntactic behavior of the sentence would have been preserved. The dependency relation is thus the one that would have been appropriate if the expression was complete, as illustrated in Figure 4. Note that while "TBC" is a parataxis, it is considered to qualify the noun "function" and was analyzed as a long form that would be inserted after the head noun "function". It is thus linked as *acl*.

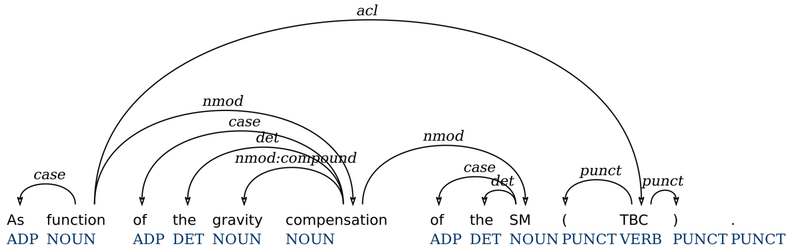

acl
nmod
case
det
nmod:compound
nmod
case
det
punct
punct
case
As function of the gravity compensation of the SM (TBC) .
ADP NOUN ADP DET NOUN NOUN ADP DET NOUN PUNCT VERB PUNCT PUNCT

Figure 4: Annotated sentence excerpt of an abbreviated short expression.

However, those abbreviations can sometimes be used as a placeholder for a number to indicate that a value has not yet officially decided. In this case, if the expression was presented in its long form, the sentence would be unnatural and even faulty. For the same reason variables' names were tagged as *NUM*, it was decided that in these cases, the POS tag should be *NUM*. This option was chosen so the POS tag would reflect the fact that it is very clear that "TBD" stands for a number and has the same role. Figure 5 illustrates this analysis.

4.3 Lists and Enumerations

One of the abundant yet problematic syntactic constructions in technical documents from the NLP point-of-view is vertical lists. Although UD guidelines specify what is the correct relation to apply within

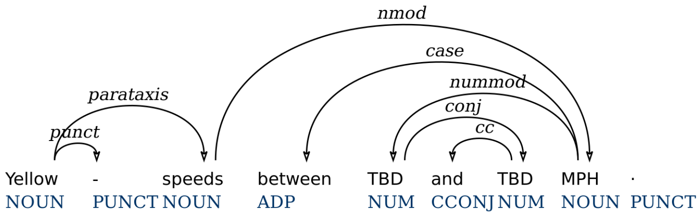
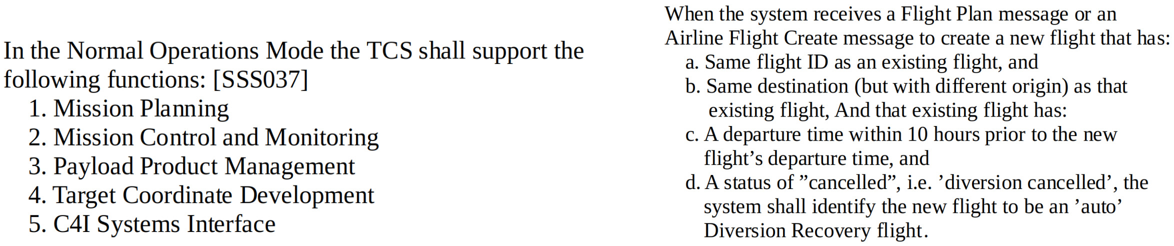


Figure 5: Annotated sentence excerpt of an abbreviation used as an undefined number.



In the Normal Operations Mode the TCS shall support the following functions: [SSS037]

1. Mission Planning
2. Mission Control and Monitoring
3. Payload Product Management
4. Target Coordinate Development
5. C4I Systems Interface

When the system receives a Flight Plan message or an Airline Flight Create message to create a new flight that has:

a. Same flight ID as an existing flight, and
b. Same destination (but with different origin) as that existing flight, And that existing flight has:
c. A departure time within 10 hours prior to the new flight's departure time, and
d. A status of ”cancelled”, i.e. 'diversion cancelled', the system shall identify the new flight to be an 'auto' Diversion Recovery flight.

(a) Full introduction.

(b) Fragmented introduction.

Figure 6: Two types of introduction sentence for list.

the elements of a list (*list*), which is only used to connect list items that do not appear in a standard syntactic construction, such as coordination, they do not specify the relation that should be used to link the introductive part of the sentence before the colons and the first element of the list.

In addition, the CoNLL-U file format does not allow line skips in sentences, nor does it allow for multiple lines of text definition in the header. This prohibits the lossless representation of sentences spanning multiple lines, as in the case of vertical lists.

4.3.1 Relation for List Introduction

Two types of syntactic structures were seen in requirements containing lists. One type uses a complete sentence as an introduction to the list, containing a verb, its subject and a complement as shown in Figure 6a. In the other type, the clause that precedes the colon is an incomplete sentence, ending with a transitive verb or any other word that needs an argument (Figure 6b).

The proposed solution for the first type of construction is to use the relation *parataxis* between the head of the clause, to which the list is linked to, and the first element of the list. For the second type of construction, however, as the part of the sentence before the colons is incomplete, the *parataxis* relation is not the best relation to use. The proposed approach (Figure 7) defines the relation between the verb (“has” in the example below) and its direct object (“ID”) as the habitual *obj*. More broadly, the relation to use is the one that would have been obvious without the colons. The resulting tree is represented below.

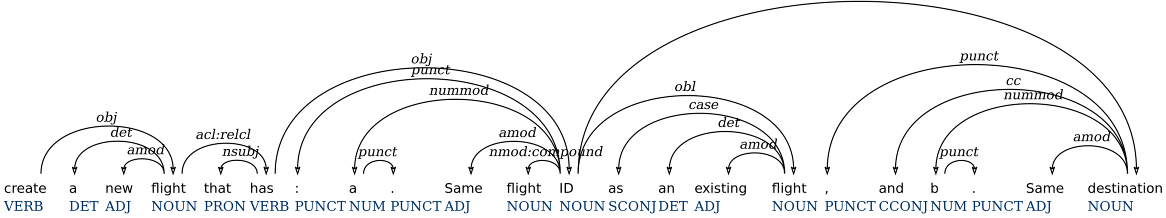


Figure 7: Example of an annotated list introduction .

4.3.2 Multilevel Lists

Multilevel lists occurred several times in the CTeTex corpus. This type of construction seems specific to technical documents. Indeed, there is no mention of them in the UD documentation and no examples were found in the other English UD corpora. However, it is easy to represent this syntactic structure with the UD relation *list*. To differentiate the different levels of lists from each other, the annotator simply has to consider them as different lists, with different heads. In the Figure 8, for example, the first list is composed of the phrases “Target CCTV” and “Device Control”, while the second list, or sublist, is composed of the words “Pan”, “Tilt” and “Zoom”. The first list then begins by “CCTV”, linked with its head “provide” with the relation *parataxis* (not with the noun “information”, as explained in Section 4.3.1). The second list begins with “Pan”, linked with its head “control” (which is the last element of the first list) with the relation *nmod*.

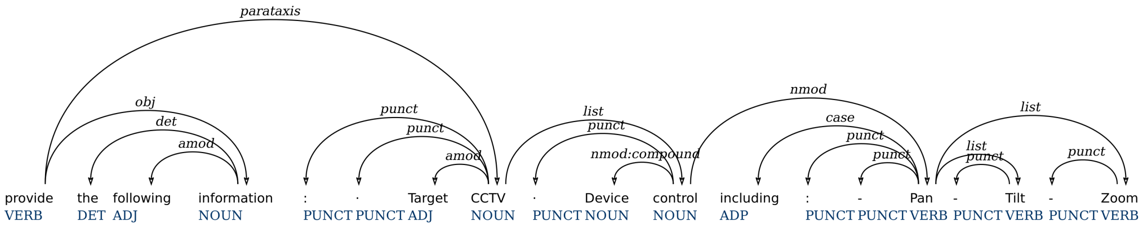


Figure 8: Example of an annotated multilevel list.

4.4 Specialized Vocabulary

Specialized vocabulary is perhaps the most obvious challenge when annotating technical documents. Annotators or readers who are not experts in the field may have trouble understanding words and multiword expressions in the text, making the annotation process longer and more complex.

4.4.1 Meta Identifiers

Some documents refer to specific SR using a unique identifier throughout the document, as shown in the example below. This identifier is sometimes situated inside the requirement (in those cases, it is usually placed after the main tensed verb of the requirement), and sometimes outside it.

> The scheduler shall [**SRS181**] set the 50 Hz interval timer to a count down value so as to cause the next minor frame interrupt at 20 msec from the previous interrupt congruently in all operational FCPs.

Here, the meta identifier *SRS181* is used to name the entire requirement. Elsewhere in the document, this requirement can be referred using this identifier. It would thus be natural to give it the POS tag *PROPN*.

The UD relation that seemingly describes this instance better is the *appos* relation. Even though the referent is not usually a noun (but rather a full sentence, thus a verb parent) as required by this relation’s definition. It also does not immediately follow its parent, as also defined in the guidelines. Nonetheless, it is proposed to extend the definition of the *appos* relation to include constructions with meta identifiers. This is because it is the only relation that really captures the function of those identifiers, which is to name the requirement. An example of the suggested analysis is represented in Figure 9.

4.4.2 Nominal Modifiers vs Compounds

Using specialized technical vocabulary makes it more difficult to differentiate between the dependency relations *nmod* (nominal modifier) and *compound*. It is probably the most influential UD guideline regarding manual annotation of technical documents, because of the wide number of cases where a decision has to be made.

At the moment, UD guidelines differentiating between compositionality and nominal modifiers are unclear: for example, in the UD documentation, “phone book” is treated as a compound (in which the

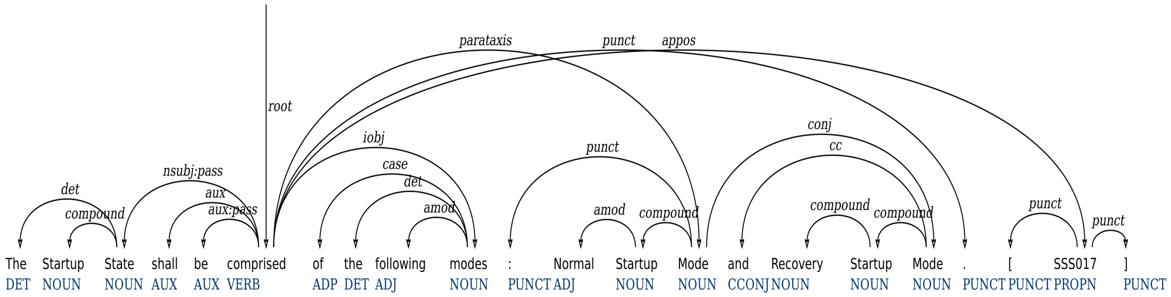

root
parataxis
punct
appos
iobj
case
det
amod
punct
conj
cc
nsubj:pass
aux
aux:pass
det
compound
amod
compound
compound
compound
punct
punct
The Startup State shall be comprised of the following modes : Normal Startup Mode and Recovery Startup Mode . [SSS017]
DET NOUN NOUN AUX AUX VERB ADP DET ADJ NOUN PUNCT ADJ NOUN NOUN CCONJ NOUN NOUN NOUN PUNCT PUNCT PROPN PUNCT

Figure 9: Example for the meta identifier identification

meaning is not compositional), as well as "ice cream flavors", of which the meaning is compositional (at least between "flavors" and "ice cream").

It is therefore proposed to follow one part of Sylvain Kahane's recommendation in this Github discussion [4]. As he suggests annotating all the Noun Noun combination with a new *nmod:compound* subrelation, we propose to use it for word group whose meaning is compositional and which could be paraphrased like (like "the dog tail" - "the tail of the dog"). This sub-relation does not yet exist, but it has the advantage of identifying borderline cases, and could easily be modified in post-processing, if a different approach is chosen.

In summary, if the nominal expression is idiomatic (non-compositional), then it is a *compound*. If it is compositional but does not possess any sort of case marking, then it is a *nmod:compound*. Finally, if it is compositional and it possesses a case marking, then it is a *nmod*.

4.4.3 Meta Qualifiers

As shown in Section 4.4.1, some requirements are accompanied by meta identifiers that are used to name the entire requirement. Similarly, some requirements are accompanied by qualifiers used to specify the status of the requirement. In the CTeTex corpus, such qualifiers indicated if the requirements were optional ("O"), mandatory ("M"), or were used as an element of information ("I") (the meaning of those letters were found directly in the document). In the following example, the requirement is qualified as mandatory.

> The network shall terminate the ongoing VCS/VBS call if it receives the 3-digit sequence "***" transmitted via DTMF signals. (M)

Those types of constructions were not found in the UD documentation. It is rather unusual to witness, in other types of documents, an element that qualifies an entire sentence instead of another word. They are the equivalent, in other technical documents, or prefixing the sentence with "It is mandatory that the systems ...", which specifies the modality of the requirement, much like an adverbial modifier. The current case play the same semantic role, but is outside of the grammatical structure of the modified sentence. While *advmod* was considered, the *parataxis* is used to link them to the root of the requirement. The relation is applied to the "M" and "O" qualifiers which are considered as having the *ADJ* POS tag, or *NOUN* for the "I" qualifier. The resulting tree is illustrated in Figure 10.

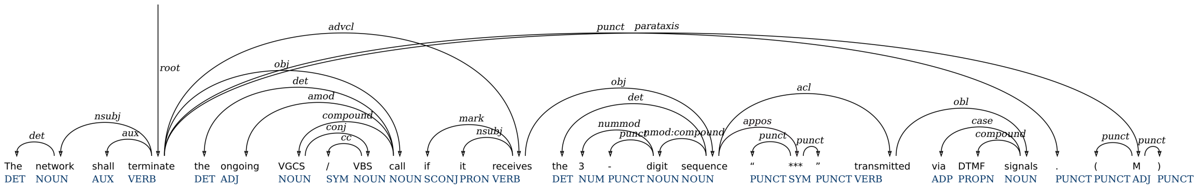

root
advcl
punct
parataxis
obj
det
amod
compound
conj
cc
mark
nsubj
obj
det
nummod
punct
nmod:compound
acl
appos
punct
punct
obl
case
compound
punct
punct
det
nsubj
aux
The network shall terminate the ongoing VGCS / VBS call if it receives the 3 - digit sequence " *** " transmitted via DTMF signals . (M)
DET NOUN AUX VERB DET ADJ NOUN SYM NOUN NOUN SCONJ PRON VERB DET NUM PUNCT NOUN NOUN PUNCT SYM PUNCT VERB ADP PROPN NOUN PUNCT PUNCT ADJ PUNCT

Figure 10: Annotated example of a meta qualifiers in a software requirement.

[4] https://github.com/UniversalDependencies/docs/issues/761

In cases where requirements are constituted of multiple sentences, it is suggested to link those indications to the head of the last sentence rather than the first one. This is to avoid bias in the statistics of sentence size and distance between tokens.

4.5 Issues with CoNLL-U Format

Other than the challenge of applying the guidelines to the CTeTex corpus, using the current CoNLL-U file format to express UD annotations is also problematic. One is that this format is lossy for multiline sentences, as there is no mechanism to express a line change within a sentence, so the resulting sentence cannot be rebuilt correctly. Another is that splitting multiline sentences into separate entries in the file will force the loss of list references as there is no cross-sentence dependency indicator in CoNLL-U format.

One way to solve these issues would be to allow intersentence parent reference with a [sentence id]-[token id] structure to avoid multiline sentences in the current format. While that would help, it would disrupt the semantics of the format by enabling the splitting of a sentence while also changing the format of the parent id. While it is feasible, it has much impact on existing resources and codebase in CoNLL-U format. A simpler solution would be to add a ”*LineAfter=Yes*” attribute in the *MISC* column to encode the line skip characters, so that it would be possible to reconstruct the exact format of the sentence. The latter option was retained to encode the CTeTex corpus as it has the least impact on the file format and it helps solve the two issues.

5 Corpora Comparison

Following the application of the guidelines, there is a need to validate if these syntactic constructs really differ from the other typical UD corpora and if they affect automatic processing tools. To that end, CTeTex is compared with other English corpora in order to view the differences in performances when a dependency parser is applied. The EWT (Silveira et al., 2014), GUM (Zeldes, 2017), GUMReddit (Behzad and Zeldes, 2020), ParTuT (Sanguinetti and Bosco, 2015), PUD (McDonald et al., 2013), LinES (Ahrenberg, 2015), Pronouns (Munro, 2021) and ESL (Berzak et al., 2016) corpora (version 2.8 of the UD dataset) serve as a basis of comparison. Some of these texts targets a specific linguistic phenomenon (Pronouns), are manually or semi-manually annotated (EWT, GUM, etc).

The left part of Table 2 shows the average sentence's length, height (or depth), arity and mean dependency distance or MDD (Jiang and Liu, 2015) for each of the eight corpora, followed by the average measure over these same corpora. The CTeTex measures are then shown, with their differences (Δ) with the actual corpora average.

The average sentence length of CTeTex if almost 50% longer and around 10% deeper than the second-highest measures from ParTUT corpus. This is expected as vertical lists often contain multiple elements that directly influence the length of the overall sentence. While arity (number of children for a node) if close to the ESL corpus, it is almost 20% higher than the average corpus. The complexity and length of sentence also influence the MDD of CTeTex which is the highest among all corpora. This indicates that tokens are less related to close neighbours that in other corpora, but are linked to parents that are often found at a greater distance in the sentence.

This might have a negative impact on automatic tools if they use a smaller contextual window to search for parent tokens. It also impacts the complexity of the annotation process, as the cognitive load of understanding complex sentence can hinder the speed of analysis.

To evaluate the influence of the nature of the texts of the proposed corpus, the nine English UD corpora were automatically annotated using Stanza v1.2.3 (Qi et al., 2020) dependency parser. Other dependency parsing tools (like Spacy (Honnibal and Montani, 2017) and UDPipe 2 (Straka, 2018)) were also tested, but produced worst overall performance on the CTeTex corpus as well as the other UD corpora. Thus only Stanza's results are presented for brevity. It should be noted that most universal dependency parsers are trained to use some version of the English UD dataset, as few other resources are publicly available. This is a methodological issue for the referred eight UD corpora as training data is usually not used for evaluation. But the hypothesis was that if CTeTEx was similar to existing UD treebanks, the difference

(Δ) between the average and CTeTex scores (for the automatic annotation columns) would have been relatively small. The delta values thus emphasis the remote nature of the content of CTeTex compared to existing annotated texts.

	Sentence (avg)				**Automatic annotation**			
Corpus	Length	Height	Arity	MDD	UPOS	UAS	LAS	CLAS
EWT	15.33	3.32	4.66	3.38	**0.9685**	**0.9345**	**0.9173**	**0.9003**
GUM	18.17	3.72	4.83	3.39	0.9543	0.8963	0.8744	0.8570
LinES	17.97	3.75	5.08	3.30	0.9246	0.8237	0.7828	0.7537
ParTUT	23.75	4.71	5.52	3.48	0.9156	0.8602	0.8038	0.7565
Pronouns	5.98	1.81	3.44	1.76	0.9599	0.8519	0.8171	0.8397
PUD	21.18	4.31	5.76	3.34	0.9586	0.8882	0.8626	0.8454
GUMReddit	18.20	3.77	5.21	3.41	0.9439	0.8585	0.8278	0.8114
ESL	19.06	3.97	5.85	3.31	0.9368	0.8999	0.8643	0.8464
Average	17.71	3.67	5.04	3.17	0.9452	0.8766	0.8437	0.8263
CTeTex	**33.60**	**5.17**	**6.01**	**3.98**	0.8699	0.7739	0.6879	0.5949
Δ	15.89	1.5	0.97	0.81	0.0753	0.1027	0.1558	0.2314

Table 2: Overview of English UD corpora compared to CTeTex. (highest values in bold)

The right section of Table 2 shows universal part-of-speech (UPOS), unlabelled association score (UAS), labelled association score (LAS) as well as the labeled association score for content words (CLAS) like nouns, verbs and adjectives. Best overall scores were obtained on the EWT corpus. One explanation might be the larger size of this resource compared to other corpora when used as training data.

While the existing corpora offer a good score for the UPOS tag, the performance on CTeTex is 7.52% lower. The reasons for such a low score might be explained by some of the decisions in Section 4 (variable as *NUM*, etc.) but also by the large number of acronyms and complex domain specific terminology. The scores continue to degrade with the addition of dependency relations (UAS), their labels (LAS) and the specific study of content words (CLAS), dropping by 23.14% from the average corpora on this last score. This goes to show that the complexity of SR texts and their underlying phenomena hinders current dependency parsers. Using information analyzed by these tools in downstream processing tasks lowers the chance of a usable outcome. The CTeTex corpus is thus a relevant contribution to improve the adaptability and stability of dependency parsers when processing technical document such as software requirements specification.

6 Conclusion

This contribution is the first English software requirements corpus annotated with Universal Dependencies part-of-speech and labelled dependency relations. The comparison to other existing corpora in English shows specificity of the CTeTex corpus as well as the challenge of processing such texts with automatic dependency annotation tools. It offers the possibility to evaluate and experiment on this type of document to improve both the UD guidelines and the automatic annotation process.

Future work on this corpus includes studying the addition of enhanced UD relations as a way to better express needed links. This would permit a better extraction of higher-level information from the requirements. Future plans also include using CTeTex to train and evaluate neural dependency parsing algorithms to improve their performance on this type of technical documents.

References

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

Imran Ahsan, Wasi Haider Butt, Mudassar Adeel Ahmed, and Muhammad Waseem Anwar. 2017. A comprehensive investigation of natural language processing techniques and tools to generate automated test cases. In *Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing*, ICC '17, New York, NY, USA. Association for Computing Machinery.

Shabnam Behzad and Amir Zeldes. 2020. A cross-genre ensemble approach to robust Reddit part of speech tagging. In *Proceedings of the 12th Web as Corpus Workshop (WAC-XII)*.

Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. Universal dependencies for learner english. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 737–746. Association for Computational Linguistics.

Alessio Ferrari, Giorgio Oronzo Spagnolo, and Stefania Gnesi. 2017. Pure: A dataset of public requirements documents. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 502–505.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Jingyang Jiang and Haitao Liu. 2015. The effects of sentence length on dependency distance, dependency direction and the implications–based on a parallel english–chinese dependency treebank. *Language Sciences*, 50:93–104.

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, June.

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

P. Ménard and S. Ratté. 2010. Classifier-based acronym extraction for business documents. *Knowledge and Information Systems*, 29:305–334.

Robert Munro. 2021. *Human-in-the-Loop Machine Learning*. Manning.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia, May. European Language Resources Association (ELRA).

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 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May. European Language Resources Association.

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*.

Michael Roth and Ewan Klein. 2015. Parsing software requirements with an ontology-based semantic role labeler. In *Proceedings of the 1st Workshop on Language and Ontologies*, London, UK, April. Association for Computational Linguistics.

M. Sanguinetti and C. Bosco. 2015. Parttut: The turin university parallel treebank. In *Italian Natural Language Processing within the PARLI Project*.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium, October. Association for Computational Linguistics.

Yinglin Wang. 2016. Automatic semantic analysis of software requirements through machine learning and ontology approach. 21:692–701, Dec.

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

Liping Zhao, Waad Alhoshan, Alessio Ferrari, Keletso J. Letsholo, Muideen A. Ajagbe, Erol-Valeriu Chioasca, and Riza T. Batista-Navarro. 2021. Natural language processing for requirements engineering: A systematic mapping study. *ACM Comput. Surv.*, 54(3), apr.

Towards a consistent annotation of nominal person in Universal Dependencies

Georg F. K. Höhn
Georg-August-University Göttingen
georg.hoehn@uni-goettingen.de

Abstract

On the basis of four small scale studies on corpora of English, German and Modern Greek, this paper points out problems with the lack of annotation guidelines for adnominal pronoun constructions like *we linguists* in treebanks employing the Universal Dependencies framework. I propose that a more uniform strategy of annotating these constructions will improve the internal consistency of corpora and better facilitate crosslinguistic comparability. Specifically, I argue against the use of the APPOS(ition) relation for these constructions and in favour of employing the DET(erminer) relation as a default annotation strategy.

1 Introduction

While nominal expressions are often assumed to be third person by default, this is not necessarily the case. Nominal person marking describes a set of phenomena where a nominal expression morphologically indicates whether its reference set contains the author and/or addressee of an utterance. Crucially, the term does not refer here to the person of a possessor. The most widely discussed type of nominal person marking are what I call adnominal pronoun constructions (APCs) like English *we linguists*.

Investigating APCs in corpora that are only POS-tagged is complicated by the fact that linear sequences of pronouns and nouns also commonly occur outside of APCs, cf. *They gave [$_{IO}$us] [$_{DO}$hope].* In languages with unambiguous case marking and given a sufficiently tagged corpus, this issue may be addressed by imposing identical case requirements on pronoun and noun, but in languages with little or no case marking on nouns (like German or English), the results of any search will inevitably contain a large number of false hits. This leads to the need for resource-intensive manual post-processing. Moreover, APCs can be syntactically complex (e.g. adjectival modifiers intervening between pronoun and noun), leading to more complex search patterns and further potential increases of false hits.

While treebanks offer an attractive way of formulating more precise search conditions to avoid a proliferation of false hits, the lack of recognition of nominal person and specifically APCs as an independent phenomenon holds back their potential in this area. I focus here on the Universal Dependencies (UD) framework (Nivre et al., 2020; de Marneffe et al., 2021)[1], showing that APCs are annotated in (at least) two divergent ways in UD corpora of German and English and in a third way in a Greek UD-corpus. This is not only undesirable because it introduces an internal inconsistency, but also impedes crosslinguistic comparability, one of the core aims of UD.

In Section 2 I sketch some theoretical and typological aspects of the phenomenon of nominal person. Section 3 describes the results of searches for APCs in English, German and Greek UD-corpora and Section 4 concludes with a proposal for a more consistent annotation of APCs in UD.

Abbreviations and glosses used: ACC = accusative, APC = adnominal pronoun construction, DEF = definite, DEM = demonstrative, DET = determiner, EXCL = exclusive, F = feminine, LIG = ligature, LOC = locative, M = masculine, N = neuter, NEG = negative, NOM = nominative, PL = plural, PRS = present, PRTCL = particle, PST = past, SG = singular, UD = Universal Dependencies.

[1] See also https://universaldependencies.org.

2 Structure and crosslinguistic variation in nominal person marking

APCs like English *we linguists* or its German (Indo-European, glottocode stan1295) counterpart (*wir Linguisten*) have been treated in the literature either as a type of apposition (Delorme and Dougherty, 1972; Olsen, 1991; Cardinaletti, 1994; Willim, 2000; Rutkowski, 2002; Ackema and Neeleman, 2013; Keizer, 2016; Ackema and Neeleman, 2018), sketched roughly in (1a), or as involving a pronominal determiner construction with the pronoun as head of a determiner phrase (Postal, 1969; Abney, 1987; Lawrenz, 1993; Lyons, 1999; Longobardi, 2008; Rauh, 2003; Roehrs, 2005; Bernstein, 2008b; Saab, 2013; Höhn, 2020), sketched in (1b).

(1) a. apposition: [NP [Pron we] [NP linguists]]

b. pronominal determiners: [DP [D we] [NP linguists]]

There is plenty of evidence against analysing English or German APCs as instances of loose apposition (Sommerstein, 1972; Pesetsky, 1978; Lawrenz, 1993; Lyons, 1999; Rauh, 2003; Roehrs, 2005; Höhn, 2016; Höhn, 2020) and most modern proponents of appositive analyses can presumably be understood in terms of close apposition (Burton-Roberts, 1975).[2]

Argumental uses of English APCs are restricted to first and second person[3] plural and they are typically in complementary distribution with the definite article, cf. *Many of us (*the) linguists are actually quite sociable*.[4] Unsurprisingly, nominal person marking crosslinguistically diverges in various ways from the English type. Closely related German, for instance, allows argumental APCs in the singular (2a), see also (Rauh, 2004). The restriction against third person adnominal pronouns is also far from universal (Höhn, 2020), compare (2b) from Hausa (Afroasiatic, glottocode haus1257). And while definite articles are excluded in regular English or German APCs, in some languages they obligatorily require a definite article, as illustrated for Greek (Indo-European, glottocode mode1248) in (2c).[5] This type of APC structure has been connected to the availability of unagreement (Choi, 2014; Höhn, 2016). Unagreement (Hurtado, 1985), illustrated in (3) for Greek, but also found, e.g., in Spanish or Bulgarian, involves a plain definite subject co-occurring with a verb inflected for first or second person (typically plural) with an interpretation largely corresponding to an APC in English.

(2) a. German

Der	Editor	ist	schon	eine	feine	Sache	für	**mich**	**Linuxer**...
DET.NOM.SG	editor	is.3SG	PRTCL	a	nice	thing	for	me.ACC	Linux.user

'The editor is quite a neat thing for me (as a) Linux user...' (attested online)[6]

b. Hausa

sū	mutā̀nê-n
they	men-DEF

'they the men' (Newman, 2000, 155)

c. Greek

opos	to	legh-ame	panta	**emis**	**i**	**vuleft-es**
like	3SG.N.ACC	say-PST.1PL	always	we.NOM	DET.PL.NOM	MP-PL.NOM

'...like we members of parliament have always said.' (UD_Greek-GDT 2049)

[2] But see Roehrs (2005) and Höhn (2020) for arguments against a close apposition analysis of English/German-type APCs as well.

[3] Some varieties/registers seem to allow argumental third plural APCs as in *them politicians*, although these pronouns have been argued to actually realise demonstratives (Maček, 1995; Rauh, 2003; Bernstein, 2008a; Hazen et al., 2011).

[4] Expressions like *we the people* have a different structure and a more restricted distribution than the *we linguists*-type APCs, see also Choi (2014, 23) and Höhn (2020, 1f.). For annotation purposes it may still be plausible to treat both as instances of nominal person marking however, see Section 4.

[5] While (2b) also contains a kind of definite marker, it is not obligatory in these constructions (Newman, 2000, 155), so Hausa and Greek probably represent different types of adnominal person marking.

[6] Archived at `https://web.archive.org/web/20211001193522/https://www.opena.tv/pc-ios-android-window-phone-programme-und-apps-fuer-stb/53395-demoneditor-fuer-linux-und-macos-post451749.html`; last accessed 1/10/2021.

(3)	me	tin	Arjentini	**i**	**Evrope-i**	ech-ume	Greek
	with	DET.ACC.SG	Argentine	DET.NOM.PL	European-NOM.PL	have-PRS.1PL	

istorik-us	dhesm-us
historical.ACC.PL	bond-ACC.PL

‘We Europeans have historical bonds with Argentine. . . ’ (UD_Greek-GDT 492)

Choi (2014) and Höhn (2016) reject an analysis of Greek-type APCs like (2c) in terms of (loose or close) apposition, see also Stavrou (1995). Choi (2014) proposes that the adnominal pronoun occupies a specifier position, see the sketch in (4a), and Höhn (2016) argues for an extension of the pronominal determiner approach along the lines of (4b). In any case, Greek APC structure clearly differs from that observed in English or German.

(4) a. [DP [DP emis] [D’ i vuleftes]]

b. [PersP [Pers emis] [DP i vuleftes]]

Adnominal pronouns/APCs are not the only means of marking nominal person, although they are the most widely attested type and this paper focuses on them. Some languages employ clitic person markers to mark nominal person in noun phrases. The Bilua (Solomon Islands, glottocode bilu1245) example in (5) illustrates a case where an adnominal pronoun and clitic person marking can co-occur. For more details on crosslinguistic variation in nominal person marking see Choi (2014), Höhn (2017) and Höhn (2020).

(5)	**enge**=a	Solomoni=a=ma	maba	poso=**ngela**	Bilua
	1PL.EXCL=LIG	Solomon=LIG=3SG.F	person	PL.M=**1PL.EXCL**	

‘we, Solomon people’ (Obata, 2003, 85, (7.35))

For current purposes, the main take-away points from this section are that a) there is no full consensus in the literature concerning the syntactic relation between the pronoun and the nominal part of an English-type APC, particularly across syntactic frameworks, and b) there is real crosslinguistic variation in the structure of nominal person marking. In the next section I will show how this is relevant to the treatment of APCs in UD corpora for English, German and Greek.

3 Prototype survey in four UD corpora

3.1 Methodology

To assess the current treatment of APCs in UD I conducted exemplary searches on the four corpora in (6) using the online tool TüNDRA (Martens, 2013).[7]

(6) a. UD_English-EWT v2.4 (Silveira et al., 2014), 251,521 tokens

b. UD_German-HDT v2.4 (Borges Völker et al., 2019), 3,399,300 tokens

c. UD_German-GSD v2.4 (McDonald et al., 2013; Bulgarian Academy of Sciences et al., 2015), 287,740 tokens

d. UD_Greek-GDT v2.4 (Prokopidis et al., 2005; Prokopidis and Papageorgiou, 2017), 61,733 tokens

UD marks syntactic relations based on the universal Stanford dependencies (de Marneffe et al., 2014). I used the relations APPOS and DET to detect APCs annotated according to one of the two main available analyses of English-type APCs in (1), as shown in the search patterns in (7ab). In order to discover the relation(s) employed for marking APCs in the Greek GDT corpus, I first searched for collocations of a first or second person pronoun adjacent to a determiner (`[pos="PRON" & person=("1"|"2")].[pos="DET"]`). The three instances of APCs found in the output were all encoded using the relation NMOD intended to be “used for nominal dependents of another noun or noun phrase and functionally corresponds to an attribute, or genitive complement”.[8] The corresponding

[7] `https://weblicht.sfs.uni-tuebingen.de/Tundra`, last accessed 4/10/2021.

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

search pattern I used to detect possible further instances of APCs is (7c).

(7) Search patterns for APCs

a. apposition (APPOS):

```
[pos="PRON" & person=("1"|"2")] >appos [pos="NOUN"]
```

b. pronominal determiner (DET):

```
[pos="NOUN"] >det [pos="PRON" & person=("1"|"2")]
```

c. nominal modifier (NMOD):

```
[pos="PRON" & person=("1"|"2")] >nmod [pos="NOUN"]
```

The following patterns were used to obtain the personal pronoun counts in English (8ab), German (8c) and Greek (8d) employed in Section 3.3. The patterns identify pronouns that can in principle occur as part of an APC, i.e. excluding possessive or reflexive pronouns. The Greek count could be further refined by excluding immediately preverbal clitic object pronouns, since that position does not allow APCs, but I refrain from doing so here.

(8) a.

```
[pos="PRON" & person="1" & number="Plur" & token!=/[o|O]ur.*/]
```

b.

```
[pos="PRON" & person="2" & token!=/[y|Y]our.*/]
```

c.

```
[token=("wir"|"Wir"|"WIR"|"uns"|"Uns"|"UNS")]
```

d.

```
[pos="PRON" & person="1" & number="Plur"] >case [pos=/.*/] |
[pos=/.*/] >!nmod [pos="PRON" & person="1" & number="Plur"]
```

3.2 Results

Table 1 lists the number of hits for each search pattern in each corpus and how many of the hits were *bona fide* APCs on manual inspection. Outside the EWT corpus, all detected APCs were 1PL. For EWT, the amount of 1PL APCs is indicated in brackets. The final two columns provide the precision and recall values for each search pattern. Recall is calculated on the assumption that the 44 APCs covered by the table exhaust the number of APCs in the corpora. Since no full manual search was conducted, there may be undetected instances of APCs which would decrease recall for all search patterns.

	EWT		**HDT**		**GSD**		**GDT**		**Overall**	
	English		*German*		*German*		*Greek*			
Pattern	Hits	APCs	Hits	APCs	Hits	APCs	Hits	APCs	**Precision**	**Recall**
(7a) APPOS	12	10 (1PL: 2)	19	14	8	5	1	0	0.725	0.659
(7b) DET	10	10 (1PL: 0)	0	0	2	2	0	0	1	0.2728
(7c) NMOD	5	0	2	0	2	0	7	3	0.1875	0.0682

Table 1: Hits and APCs among hits per corpus by search patterns, overall precision and recall for each search pattern

The English EWT and German GSD corpora encode APCs inconsistently as apposition or pronominal determiners, while the German HDT corpus exclusively employs the APPOS relation and Greek GDT only the NMOD relation. While the DET relation was only used for APCs in the EWT and GSD corpora, partly accounting for the relatively low recall value, it showed the highest precision where employed. In all corpora the apposition relation also included hits that were not APCs like German (9), reflected in a lowered precision value. Considering the relative flexibility of the notion of apposition this is not surprising. Some of these hits involved apposition to pronouns as indicated by commas or parentheses, like English (9).

(9)

“Wenn	wir	dem	Konsumenten	Atmosphäre	verkaufen,	sind	**wir**	German
if	we	DET.DAT.SG	consumer	atmosphere	sell.1/3PL	are.1/3PL	we	

die	ersten	Ansprechpartner,	nicht	die	**Illegalen**”	…
DET.NOM.PL	first	contact.person.PL	not	DET.NOM.PL	illegal.PL	

“If we sell the consumer atmosphere, we will be the first point of call, not the illegal ones (sources for downloading music).” (UD_German-HDT 12632)

(10) a. Also, can animals remember images on TV like **us**, **humans**? (UD_English-EWT 12553)

b. I ca nt [sic!] speak for them but any tests or appointments they recommend are probably in the best interests of **us** (the **patient** [sic!]) and … (UD_English-EWT 15813)

The NMOD relation in (7c) picked out no hits in the Greek corpus beyond those identified by the linear search pattern used to identify the NMOD relation as described above, including (2c). The pattern also yielded several non-APC results like the PP modifier in (11), as reflected by the low precision value in Table 1. There were no APC matches for the pattern in any of the other corpora.

(11)

…ochi	mono	se	**emas**	sto	**Kinovulio**…	Greek
NEG	only	LOC	we.ACC	LOC.DET	Parliament	

‘not only to us in the Parliament’ (UD_Greek-GDT 2100)

Overall, these observations illustrate that not only does the current annotation practice not provide a crosslinguistically consistent means of identifying APCs, but it also lacks consistency within languages.

3.3 Differences in frequency

While this paper has a methodological focus, a brief comparison of the results by corpus may be instructive as a starting point for future research. As a basic comparative measure of the prevalence of APCs in a given corpus, Table 2 indicates the frequency of APC relative to the number of pronouns with the same person/number combination in a corpus.[9]

	EWT		**HDT**	**GSD**	**GDT**
	English		*German*	*German*	*Greek*
	1PL	2PL	1PL	1PL	1PL
# pronouns	1334	(2771)	3012	441	89
# APCs	2	18	14	7	3
Freq (APCs)	0.15%	(0.65%)	0.465%	1.587%	3.371%

Table 2: Relative frequencies of APCs

These data display clear contrasts in the relative frequency of APCs between the corpora. The fact that, in spite of their marked difference from each other, both German corpora have a markedly higher 1PL APC frequency than the English EWT might suggest that language-level differences have at least some role to play in addition to other factors (e.g. genre or speech style).

Of course, due to the limited size of the datasets this can only serve as a first tentative approximation to the issue. Particularly for the Greek GDT corpus, the small corpus size and low number of 1PL pronouns prevents any strong claims for now. If, however, the comparatively high frequency for Greek APCs turns out to be corroborated by more data, this might be connected to the generally marked nature of overt pronouns in pro-drop languages on the one hand and the availability of unagreement, cf. (3), on the other hand.

[9] Note that the frequency for 2PL APCs in English is only provided for rough orientation and is not directly comparable the other values because the number-ambiguity of *you* the # pronouns cell includes singular and plural uses in this case. The real 2PL APC frequency in English must be higher than indicated.

More data and closer investigation of further potential parameters (e.g. genre, style, modality) are needed to establish which factors influence the frequency of APCs and to clarify whether there are stable language-level differences, but these preliminary observations show the potential role of corpus research in a better understanding of the crosslinguistic distribution of nominal person phenomena.

4 Discussion

The inconsistent treatment of APCs observed above is a clear shortcoming and harmonisation, at least within the same language, is highly desirable. So how can the situation be improved?

To start, encoding English or German APCs with the UD-relation APPOS seems questionable, even setting aside the question of the most appropriate theoretical analysis. The UD documentation explicitly aims at employing the APPOS relation for loose apposition, noting that "the two halves of an apposition can be switched".[10] This diagnostic does not apply to English, German or Greek APCs (12) and as mentioned in Section 2 the literature widely rejects a loose apposition analysis for these languages.[11]

(12) a. *linguists we

b. *Linguisten wir — German
linguists we

c. *i ghlosoloj-i emis — Greek
DET.NOM.PL linguist-NOM.PL we.NOM

Against this background, one way of dealing with English- and German-type APCs is to systematically employ the DET relation. This should maintain consistency with the current definitions of relations and requires only minor clarifications to the guidelines. While the recall rates in Table 1 for APPOS are considerably higher than for DET, this is of course mainly a reflection of it being the more widely (albeit not consistently) used annotation strategy in the corpora. The lower precision for the APPOS search pattern is, on the other hand, systematically inevitable precisely because that relation is also used for constructions that are clearly not akin to APCs, e.g. (9) and (11).

Concerning appositives with pronoun-noun collocations like (10), the typographic convention of using commas in languages like English or German permits a distinction from APCs in written corpora with a certain amount of confidence. As correctly implied by a reviewer, this convention may not be followed consistently, especially in informal writing (e.g. web corpora) and it is unlikely to be helpful in corpora on less or non-standardised languages. However, this does not mean that such typographic cues (and theoretical insights) should be ignored where available. There may be further language-specific indicators for identifying certain pronoun-noun collocations as real cases of apposition. In the absence of such indications, however, I advocate against using the APPOS relation as a default. While annotation inconsistencies are bound to occur in any case, using the APPOS relation for APCs, unclear cases of APCs and various other constructions systematically reduces the precision rate for searches for APCs as pointed out above. Using the DET relation as default instead allows keeping "potential" APCs apart from other constructions. Closer inspection of the "unclear" APC cases may in turn enable the discovery of more of the abovementioned "further language-specific indicators" of apposition vs. APCs.

How should we pursue the aim of crosslinguistic comparability in the face of the observable variation? Just like APPOS is not a satisfactory label for the relation between adnominal pronouns and their nominal complement in English-type APCs, I do not think the NMOD relation currently employed in the Greek corpus is an attractive solution for encoding APCs in that type of language. It again conflates APCs with different, unrelated constructions (PP modifiers, genitives), does not contribute to the goal of crosslinguistic comparability and obfuscates the fact that even though English and Greek APCs have clear structural differences, they are comparable at least on a descriptive or phenomenological level.

Since UD does not impose a limit on the number of DET relations with a noun,[12] both the adnominal pronoun and the article in Greek APCs like (2c) may be analysed with a DET relation from the noun.

[10] https://universaldependencies.org/u/dep/appos.html

[11] But see Höhn (2017, 46–50) for a small number of languages for whose APCs this diagnostic may apply after all.

[12] https://universaldependencies.org/u/pos/DET.html

This is consistent with UD's relatively broad notion of determiners as "express[ing] the reference of the noun phrase in context" and parallels the annotation of Greek demonstrative constructions like (13) in the GDT corpus, where both the demonstrative and the article are in a DET relationship to the noun. A structural parallelism between demonstrative modifiers and adnominal pronouns in languages with Greek-type APCs has been independently argued for by Choi (2014) and Höhn (2016) respectively, although the specifics of their analyses differ from the relational model of UD.

(13)	aft-i	i	stochi	Greek
	DEM-NOM.PL	DET.NOM.PL	goal-NOM.PL	
	'these goals'			(UD_Greek-GDT 88)

This brings me to a question raised by a reviewer concerning the treatment of English constructions like *we the people*. That construction has a more restricted distribution than English APCs (Höhn, 2020, 1f.), so while it is arguably an expression of nominal person, it is distinct from regular English APCs. Since the definite article makes the distinction easily detectable, marking this construction using two DET relations in UD seems to be a plausible practical approach for descriptive purposes. I should stress that I consider this parallel to Greek-type APCs to be a surface similarity only, with deeper syntactic (and distributional) differences between Greek-type APCs and English *we the people* best addressed within a more fine-grained theoretical framework.

Two reviewers point out that the core data discussed here stem from a relatively restricted range of languages. I agree that this limits possible conclusions concerning crosslinguistic variation in APCs, but this is not the focus of this paper – see Höhn (2017) for a first larger scale view of variation in nominal person. My point here is that even among three relatively well-documented languages there is no consistent annotation of APCs (or alternative means of nominal person marking).

Based on Höhn (2017), APCs of different sorts (main points of crosslinguistic variation: pre- or postnominal pronoun, with or without article, can APCs combine with demonstrative modifiers or not) represent the most common class of expressions of nominal person (found in 74 of 87 investigated languages). My impression is that these types of APCs should be effectively analysable in UD using the DET relation along the lines suggested above, even though they may involve different types of underlying syntactic structures on closer inspection in different syntactic frameworks. Matters are not quite as straightforward for languages that employ clitic person markers instead of or alongside additional person-sensitive[13] determiners, cf. the Bilua example in (5) and Höhn (2017, Ch. 2.3.4). It is possible that nominal person in these languages could still be encoded using the DET relation, but one would have to decide which word class the clitic markers should be assigned to if annotated as separate words (possibly similar to clitic pronouns in other languages). Alternatively, one may have to permit PERSON as a nominal inflectional feature. While I leave this issue open here, the APPOS relation would seem generally inapplicable here in any case.

5 Conclusion

I have shown that APCs, one type of nominal person marking, are currently inconsistently annotated in UD-annotated corpora and argue in favour of transparent, internally – and as far as possible also crosslinguistically – consistent guidelines. Specifically, I propose to avoid the use of the APPOS or NMOD relations to capture typical APCs. While there may be need for further language-specific guidelines in some languages, the systematic annotation of APCs using the DET relation seems to be a good practical strategy for a large number of languages, even in cases where APCs involve another determiner.

Transparent and consistent guidelines for the treatment of APCs and eventually also other expressions of nominal person will allow UD-based corpora provide a solid empirical basis for comparative investigations concerning their use and frequency. Beyond the theoretical interest, taking seriously the annotation of expressions of nominal person should also benefit applications like machine translation, as it may facilitate a more straightforward generation of translation equivalents for unagreement constructions like (3), which currently seem to pose a challenge for automatic translation.

[13] These determiners do not always correspond to full pronouns, cf. Khoekhoe/Nama (Haacke, 1977).

References

Steven Abney. 1987. *The English Noun Phrase in its Sentential Aspect*. Ph.D. thesis, MIT.

Peter Ackema and Ad Neeleman. 2013. Subset controllers in agreement relations. *Morphology*, 23:291–323.

Peter Ackema and Ad Neeleman. 2018. *Features of Person*. MIT Press, Cambridge (MA).

Judy B. Bernstein. 2008a. English *th*-forms. In Henrik Høeg Müller and Alex Klinge, editors, *Essays on Nominal Determination: From morphology to discourse management*, pages 213–232. John Benjamins, Amsterdam.

Judy B. Bernstein. 2008b. Reformulating the determiner phrase analysis. *Language and Linguistics Compass*, 2(6):1246–1270.

Emmanuel Borges Völker, Maximilan 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)*, Paris.

Bulgarian Academy of Sciences, Eberhard-Karls-Universität, Copenhagen Business School, Danish Society for Language and Literature, University of Groningen, Universität Potsdam, Universität des Saarlandes, Universität Stuttgart, Eberhard-Karls-Universität Tübingen, University of Southern Denmark, SINTEF Telcom & Informatics, Jožef Stefan Institute, Charles University, The Fran Ramovš Institute for the Slovenian Language, University of Barcelona, Uppsala University, Växjŏ University, and Middle East Technical University. 2015. 2006 conll shared task - ten languages.

Noel Burton-Roberts. 1975. Nominal apposition. *Foundations of Language*, 13:391–419.

Anna Cardinaletti. 1994. On the internal structure of pronominal DPs. *The Linguistic Review*, 11:195–219.

Jaehoon Choi. 2014. *Pronoun-Noun Constructions and the Syntax of DP*. Ph.D. thesis, University of Arizona.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford Dependencies: A cross-linguistic typology. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC2014)*, Reykjavik. European Language Resources Association (ELRA).

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

Evelyn Delorme and Ray C. Dougherty. 1972. Appositive NP constructions. *Foundations of Language*, 8:2–29.

Wilfrid Heinrich Gerhard Haacke. 1977. The so-called "personal pronoun" in Nama. In Anthony Traill, editor, *Khoisan Linguistic Studies 3*, pages 43–62. African Studies Institute, Johannesburg.

Kirk Hazen, Sarah Hamilton, and Sarah Vacovsky. 2011. The fall of demonstrative them. Evidence from Appalachia. *English World-Wide*, 32(1):74–103.

Alfredo Hurtado. 1985. The unagreement hypothesis. In L. King and C. Maley, editors, *Selected Papers from the Thirteenth Linguistic Symposium on Romance Languages*, pages 187–211, Amsterdam. John Benjamins.

Georg F. K. Höhn. 2016. Unagreement is an illusion: Apparent person mismatches and nominal structure. *Natural Language and Linguistic Theory*, 34(2):543–592.

Georg F. K. Höhn. 2017. *Non-possessive person in the nominal domain*. Ph.D. thesis, University of Cambridge.

Georg F. K. Höhn. 2020. The third person gap in adnominal pronoun constructions. *Glossa: a journal of general linguistics*, 5(1):69.

Evelien Keizer. 2016. We teachers, you fools: Pro+n(p) constructions in functional discourse grammar. *Language Sciences*, 53:177–192.

Birgit Lawrenz. 1993. *Apposition. Begriffsbestimmung und syntaktischer Status*. Narr, Tübingen.

Guiseppe Longobardi. 2008. Reference to individuals, person, and the variety of mapping parameters. In Henrik Høeg Müller and Alex Klinge, editors, *Essays on Nominal Determination: From morphology to discourse management*, pages 189–211. John Benjamins, Amsterdam.

Christopher Lyons. 1999. *Definiteness*. Cambridge University Press, Cambridge.

Scott Martens. 2013. Tündra: A web application for treebank search and visualization. In *Proceedings of The Twelth Workshop on Treebanks and Linguistic Theories (TLT12)*, pages 133–144, Sofia.

Dora Maček. 1995. The development and function of the dialectal them. *Studia Romanica et Anglica Zagrabiensia*, 40:221–235.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Tackstrom, Claudia Bedini, Nuria Bertomeu Castello, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of ACL 2013*.

Paul Newman. 2000. *The Hausa Language. An Encyclopedic Reference Grammar*. Yale University Press, New Haven, London.

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 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4034–4043, Marseille. European Language Resources Association.

Kazuko Obata. 2003. *A grammar of Bilua. A Papuan language of the Solomon Islands*. Number 540 in Pacific Linguistics. The Australian National University, Canberra.

Susan Olsen. 1991. Die deutsche Nominalphrase als Determinansphrase. In *DET, COMP und INFL: Zur Syntax funktionaler Kategorien und grammatischer Funktionen*, pages 35–56. Niemeyer, Tübingen.

David Pesetsky. 1978. Category switching and so-called so-called pronouns. In Donka Farkas, Wesley M. Jacobsen, and Karol W. Todrys, editors, *Chicago Linguistic Society*, volume 14, pages 350–360, Chicago.

Paul Postal. 1969. On so-called "pronouns" in English. In David A. Reibel and Sanford A. Schane, editors, *Modern Studies in English: Readings in Transformational Grammar*, pages 201–226. Prentice Hall, Englewood Cliffs (New Jersey).

Prokopis Prokopidis and Haris Papageorgiou. 2017. Universal dependencies for Greek. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 102–106, Gothenburg.

Prokopis Prokopidis, Elina Desypri, Maria Koutsombogera, Haris Papageorgiou, and Stelios Piperidis. 2005. Theoretical and practical issues in the construction of a Greek dependency treebank. In Montserrat Civit, Sandra Kubler, and Ma. Antonia Marti, editors, *Proceedings of The Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, pages 149–160, Barcelona.

Gisa Rauh. 2003. Warum wir Linguisten "euch Linguisten", aber nicht "sie Linguisten" akzeptieren können. Eine personendeiktische Erklärung. *Linguistische Berichte*, 196:390–424.

Gisa Rauh. 2004. Warum 'Linguist' in 'ich/du Linguist' kein Schimpfwort sein muß. Eine konversationstheoretische Erklärung. *Linguistische Berichte*, 197:77–105.

Dorian Roehrs. 2005. Pronouns are determiners after all. In Marcel den Dikken and Christina M. Tortora, editors, *The Function of Function Words And Functional Categories*, pages 251–285. John Benjamins, Amsterdam.

Pawel Rutkowski. 2002. Noun/pronoun asymmetries: evidence in support of the DP hypothesis in polish. *Jezikoslovlje*, 3(1–2):159–170.

Andrés Saab. 2013. Anticoncordancia y sincretismo en Español. Unagreement and syncretism in Spanish. *Lingüística*, 29(2):191–229.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Alan H. Sommerstein. 1972. On the so-called definite article in English. *Linguistic Inquiry*, 3:197–209.

Melita Stavrou. 1995. Epexegesis vs. apposition in Modern Greek. In *Scientific Bulletin of the School of Philology*, volume 5, pages 217–250. Aristotle University, Thessaloniki.

Ewa Willim. 2000. On the grammar of Polish nominals. In Roger Martin, David Michaels, and Juan Uriagereka, editors, *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, pages 319–343. MIT Press, Cambridge (MA).

UDWiki: guided creation and exploitation of UD treebanks

Maarten Janssen
Charles University
Faculty of Mathematics and Physics
Prague, Czechia
janssen@ufal.mff.cuni.cz

Abstract

UDWiki is an online environment designed to make creating new UD treebanks easier. It helps in setting up all the necessary data needed for a new treebank up in a GUI, where the interface takes care of guiding you through all the descriptive files needed, adding new texts to your corpus, and helping in annotating the texts. The system is built on top of the TEITOK corpus environment, using an XML based version of UD annotation, where dependencies can be combined with various other types of annotations. UDWiki can run all the necessary or helpful scripts (taggers, parsers, validators) via the interface. It also makes treebanks under development directly searchable, and can be used to maintain or search existing UD treebanks.

1 Introduction

There are many tools available to work with Universal Dependency (UD) - to annotate, visualize, edit, or processes treebanks. The majority of those tools are components only handling part of the construction of a treebank. This makes it necessary for people interested in creating a new treebank to read up on quite a few things about UD: on the principles behind the framework, on which documents are needed to submit a treebank, on which features would apply to the language in question, etc. Given the large amount of languages for which there is a UD treebank, all this is clearly not an insurmountable hurdle. But it likely makes that only people with a more computationally oriented profile will be attracted to create a treebank. And for less resourced languages (LRL), there might not always be computationally oriented linguists available.

This paper describes a more global environment for creating and maintaining UD treebanks, which aims to bring all the steps needed to create a new treebank into a single location, and which tries to make the process as easy as possible. This environment, called UDWiki, is a re-implementation of the by now defunct CorpusWiki (Janssen, 2016a), which was an attempt to provide a guided interface to allow people to build their own Part-of-Speech (POS) tagged corpus.

UDWIKI is built on top of the TEITOK (Janssen, 2016b) online corpus environment. TEITOK provides all the necessary tools to maintain (UD) treebanks, and UDWiki adds several specific tools to create treebanks for new languages from scratch, and make sure the treebank adheres to the UD standards. In its current state, UDWiki is fully operational, and has been tested on actual treebanks. But it is not yet a polished model - the idea behind the system is to fine-tine it based on the needs arising from the actual use of the system.

This paper describes the philosophy behind UDWiki and its implementation in TEITOK. It will describe UDWiki from the perspective it was initially conceived for - the creation of treebanks for new languages. And due to feedback from the reviewers it will also describe TEITOK as a generic tool for working with UD treebanks, and highlight the additional options the TEITOK environment brings over CoNLL-U based treebanks - mostly concerning mark-up that go beyond base UD annotation.

2 TEITOK and UD

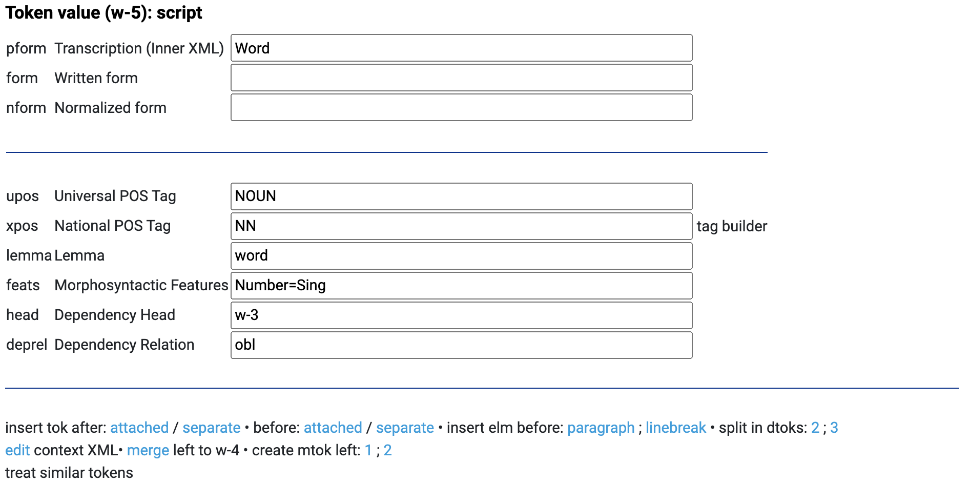
TEITOK is an online environment for building, maintaining, publishing and searching linguistically annotated corpora. It was not initially designed for UD, and does not use the CoNLL-U format for its internal storage. It rather stores corpus files in the TEI/XML format, where UD information can be embedded. TEITOK has incorporated several specific tool to work with UD treebanks (Janssen, 2018), and is used for the publication of a full version of the UD treebanks at LINDAT[1].

Dependency syntax in TEI is often represented in a manner quite unlike CoNLL-U, in which the morphology and lemma is kept on the word, while the dependency relations are kept as stand-off links elsewhere in the XML. An examples can be found in the SSJ500k corpus (Dobrovoljc et al., 2019). The approach in TEITOK is less strictly TEI, but much closer to CoNLL-U, and can be seen as a direct XML version of the CoNLL-U format: all tokens are modeled as ⟨tok⟩ nodes, the original text (form) is kept as the text content of that node, and all other columns in a CoNLL-U file are kept as attributes. So the difference in representation of tokens between TEITOK and CoNLL-U is small: instead of lines with columns there are nodes with attributes. An example of an annotated token in TEITOK is given in Figure 2, where the first 9 attributes correspond to the various columns in CoNLL-U, while the last two attributes (@id and @head) are specific to the TEITOK format. The two empty attributes (@deps and @misc) in this example are kept for clarity, but in TEITOK empty attributes would typically be suppressed.

```
<tok ord="3" lemma="word" upos="NOUN" xpos="NN" feats="Number=Sing"
  uhead="5" deprel="obl" deps="_" misc="_"
  id="w-5" head="w-3">Word</tok>
```

Figure 1: A Token in the TEITOK/XML representation

TEITOK is meant not only to publish, but also edit corpora. The editing of tokens is done in an intuitive way: from the running text, just click on a word. This will pop-up an HTML form as in figure 2, where all the attributes of the token can be directly corrected.



Token value (w-5): script

pform	Transcription (Inner XML)	Word	
form	Written form		
nform	Normalized form		
upos	Universal POS Tag	NOUN	
xpos	National POS Tag	NN	tag builder
lemma	Lemma	word	
feats	Morphosyntactic Features	Number=Sing	
head	Dependency Head	w-3	
deprel	Dependency Relation	obl	

insert tok after: attached / separate • before: attached / separate • insert elm before: paragraph ; linebreak • split in dtoks: 2 ; 3
edit context XML• merge left to w-4 • create mtok left: 1 ; 2
treat similar tokens

Figure 2: The Token Editor in TEITOK

So the sequence of ⟨tok⟩ nodes in the TEITOK document corresponds very directly with the sequence of lines in a CoNLL-U file. But all the other information in TEITOK is modeled quite differently. Generally speaking, a TEITOK/XML document can contain more information than a CoNLL-U file, and in a more structured fashion, for a number of reasons.

Firstly, there is no limit to the amount of attributes a (token) node can hold. And in TEITOK, you define for each corpus which attributes should be used. The example in figure **??** contains attributes relating to UD, but in a different corpus tokens can have very different attributes, some examples of which will be given in section 2.1.

[1] http://lindat.mff.cuni.cz/services/teitok/ud27/index.php

The fact that we can add attributes easily takes away the need to overload the `@misc` or other columns in UD with more and more information, since new types of information can get their own attributes. There are of course tabular formats where this is also possible, such as the CoNLL-U plus (conllup) format, where arbitrary numbers of columns can be specified. But for the representation of data with variable amounts of columns, XML tends to be a more amicable format. For instance, if we have attributes that appear only a handful of times in a large corpus, a tabular format like conllup forces you to add that attribute to all tokens, and be empty in the majority of them. In XML you can simply use the attribute only where needed. So new attributes can be added when necessary, without having to change anything in the existing files.

Secondly, everything in CoNLL-U that is not a token line is treated as a comment, with a largely loose convention of naming and content. Whereas in TEITOK, sentences and paragraphs are structural nodes, which just like tokens can be adorned with various types of annotations. This can be used for translation glosses as is found in CoNLL-U, but also for instance for phrasal typology, or for discourse relations defining dependencies between sentences rather than tokens. And structural elements are not limited to paragraphs and sentences - it is also possible to demarcate utterances, verse lines, chapters, and other types of segmentation of the text, each with their own set of attributes.

Thirdly, TEITOK can contain segments that should not be exported to the corpus. This is useful when the corpus contains parts that are not representative for the language. For instance, older or LRL text are not infrequently interspersed with text in another language - say Latin or English. And a typical format for LRL corpora is a non-native speaker interviewing a native speaker of the language. In both of those cases, the foreign or non-native parts should not be seen as representative of the language and hence should not be included in the treebank. But the content becomes uninterpretable by leaving those passages out. Therefore there are various ways in TEITOK to incorporate content that is not to be exported to the corpus, using for instance the ⟨`foreign`⟩ tag.

And finally, where CoNLL-U files typically contain little in the way to document metadata, in TEITOK each document is a separate TEI/XML file, with a full metadata header (teiHeader) that can specify almost anything about the text, such as for instance metadata about dialect, date, social status of the author, etc. And for many types of linguistic research, such metadata are often of crucial importance when working with corpus data. So when using a treebank for anything else than pure syntactic relations, it is vitally important to keep track of metadata. In general, TEITOK does not pose any restrictions on metadata, so the correct use in terms of representation, interpretations, and consistency is up to the corpus administrator. But for a specific corpus, or a collection of corpora such as in UDWiki, the obligatory metadata fields can be defined explicitly, and can be limited to fixed value lists to enhance consistency.

The idea behind TEITOK is not to modify the UD representation, which is why it uses the UD columns verbatim. But as shown above, the XML format offers the possibility to represent information differently, and makes it easier to incorporate new types of information as is often desired in treebanks.

2.1 Markup beyond UD

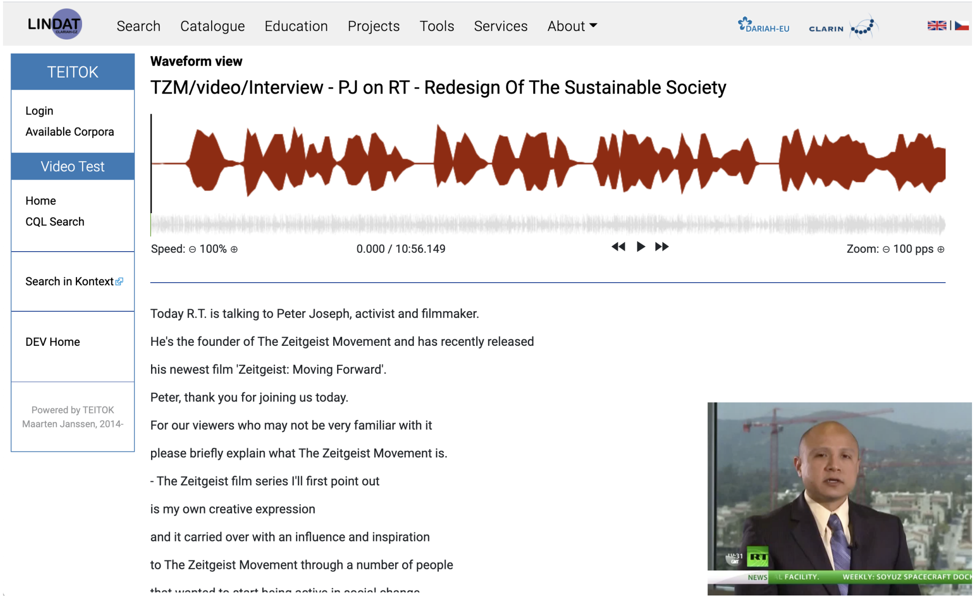
In TEITOK, the tokenization is not done over clean text, as is customary in traditional NLP, but rather over full TEI/XML documents, which can contain any type of TEI markup. This is often very important when working with anything but modern printed material: manuscript-based corpora will have added and deleted elements, unreadable and supplied element, and changes of hand, etc. Spoken corpora contain pauses, repetitions, truncations, etc. For the proper interpretation of the corpus, all those elements are often important, and flattening them out can lead to incorrect conclusions. There are UD treebanks that keep this type of information, such as for instance the spoken-specific data in the Slovenian Spoken Treebank (Dobrovoljc and Nivre, 2016), but those use largely ad-hoc solutions with limited expressive power. TEI offers a standardized representation for spoken phenomena (as well as many other types of (meta)linguistic information), which has been used, tested, and refined in a large number of projects, and offers a rich representation language.

The additional textual mark-up in TEI can also be used for stylistic information: headers, footers, bold, italics, small-caps, etc. All of these tend to have linguistic consequences, and when possible it is much better to keep them. Keeping typesetting information makes it possible to read the entire document in a pleasant way. For modern printed English texts, that is not too important, but UDWiki is explicitly targeted towards LRL data - and for many of those languages, the treebank might be one of the few available resources online. By making the documents properly typeset, the treebank can serve as a collection of textual resources for the language community outside of the academic realm. By inserting ⟨tok⟩ nodes inside the existing TEI document, TEITOK keeps all this information, in a manner that does not interfere with the UD annotations over tokens, something that is hard to do in a tabular set-up.

As mentioned in the previous section, TEITOK documents can include any number of attributes, and not only fields to annotate dependency relations and morphosyntax. For instance, TEITOK can have (multiple) normalized orthographies for tokens. This is important in historical corpora, LRL corpora and chat-style corpora where there often is a lot of orthographic variation. Normalized orthography is kept explicitly as an attribute over the original content. Providing a normalized orthography enhances searchability. And both the original orthography and the normalized spelling(s) can be made searchable.

Additional attributes can also be used for semantic frames such as the ValLex frame in PDT, as is done in the TEITOK version of PDT-C[2]. And it has been used for several types of explicit codifications, such as an error code in learner texts to mark out deviations from the native norm, or linguistic codes to mark dialect-specific features in a text.

TEITOK documents can also have associated audio files, and furthermore have time-alignment between the utterances and the audio, making it possible to listen to the part of the audio corresponding to the utterance directly. And similarly, it can have facsimile image for historical corpora based on manuscript transcription, aligned not only with each page of the text, but also with each line or each word. Both of these options make it possible to directly verify the source material in case of potential transcription errors. An example of how all this information is exploited in TEITOK is given in figure 3, which shows an audio track from a video (shown on the bottom) with a transcription that scrolls the current utterance into view while playing the audio. This example was created from a subtitle (srt) file, and does not contain any speech-specific markup (see Janssen (2021) for examples with full speech markup). Crucially, this example is generated from a TEITOK/XML file that contains the full dependency parsing information in UD format.



Waveform view

TZM/video/Interview - PJ on RT - Redesign Of The Sustainable Society

Today R.T. is talking to Peter Joseph, activist and filmmaker.
He's the founder of The Zeitgeist Movement and has recently released
his newest film 'Zeitgeist: Moving Forward'.
Peter, thank you for joining us today.
For our viewers who may not be very familiar with it
please briefly explain what The Zeitgeist Movement is.
- The Zeitgeist film series I'll first point out
is my own creative expression
and it carried over with an influence and inspiration
to The Zeitgeist Movement through a number of people
that wanted to start being active in social change

Figure 3: An example of an aligned video in TEITOK

TEITOK/XML files can also incorporate additional types of annotations, such as named entities with their respective classifications and linkings, quotations, references, footnotes, morphological decompo-

[2] https://lindat.mff.cuni.cz/services/teitok/pdtc/index.php

sitions, etc. So generally speaking, a treebank in TEITOK can be richer than what CoNLL-U supports. Of course, if TEITOK is used to create a treebank that is to be exported to the CoNLL-U format, much of this additional information will be lost in the export. But the TEITOK/XML files themselves do contain all the information, which can be exploited in a variety of different ways, while not hampering the representation of the core UD information. And the additional information is store in a structural way, so that if additional features are added to the UD format, it is just a matter of exporting that information in the appropriate way.

2.2 Searching

Searching in TEITOK is not done directly in the TEI/XML files, but rather by first loading the corpus files into some selected corpus query system (CQS). The CQS is used as an index over the corpus: a query is sent in the language of the CQS to the query system, which then returns a combination of the document ID and the token ID of the results. TEITOK then converts those results to XML fragments, and displays those fragments together with a link to the full original context. This means that all the information present in the original XML, including things that were not or cannot be exported to the CWB corpus, are present and visible in the search result. Since TEITOK has a modular design, the corpus can be made available via various CQS.

The default CQS in TEITOK is the Corpus WorkBench (CWB) (Evert and Hardie, 2011), which allows searches in the Corpus Query Langauge (CQL). When exporting to CWB, TEITOK also generates an index linking token IDs to byte-offsets in the original XML files. And with the method described above, the CQL results are rendered as XML fragments. The byte-offset index created for the CWB is also used to convert the results of any other CQS to XML fragments in an efficient manner.

CQL cannot (really) search in dependency trees, only in sequences of tokens, which is not ideal for UD treebanks. For the publication of UD2.7 in TEITOK, we therefore added the option at LINDAT to not only export the corpus to CWB (and Kontext[3], see Janssen (2021)), but also to PML-TQ (Pajas et al., 2009). PML-TQ is a query language meant explicitly to search dependency parsed corpora. And for the UDWiki project, a search module using Grew Match (Guillaume, 2019) has been added as well. The grew search works slightly differently, since Grew searches directly in CoNLL-U files. So each XML file in TEITOK is exported to a separate CoNLL-U file, with the token ID encoded in the `misc` column, which makes grew match results provide both the filename and the token ID (see also section 2.3).

SAll corpora in TEITOK can be updated upon request at any point, by running the indexing script from the interface. In that fashion, UD treebanks in TEITOK can be made searchable in all CQS after every correction or extension to the corpus. This means that the the corpus can become searchable directly, rather than having to wait for the next release of UD. For developers of the corpus this allows them to use the query language to search for possible errors in their corpus, by means of a range of query languages. And the result is linked directly to the XML file, which can be easily edited in TEITOK. For CQL, TEITOK even offers the option to edit directly from the KWIC results, which tends to be very helpful in known structural errors. For instance, the Spanish *que* can be either a relative pronoun, or a subordinating conjunction. And many taggers are surprisingly bad at distinguishing between the two. Such structural errors can be efficiently corrected by searching for constructions where we know the tagger gets it wrong, say the word *que* of which the head is a main verb that is not in the subjunctive, and then correcting the results - either by changing all of them in one go, or by manually checking them one by one.

2.3 Comparison

In order to get a good idea of the status of TEITOK as a tool for UD, it is important to compare it to other environments used to work with UD. On the one hand, there are dedicated tools to work with UD treebanks in CoNLL-U format, such as ConlluEditor (Heinecke, 2019) and Grew (Guillaume, 2019). And on the other hand there are generic tools that can be used for UD treebanks, the most popular

[3] `https://github.com/czcorpus/kontext`

amongst which are stand-off annotation tools like BRAT (Stenetorp et al., 2012) and ANNIS (Krause and Zeldes, 2016).

If we abstract away from the representation format, the interface of ConlluEditor is different from the tree editing mode in TEITOK, but the functionality is largely similar. However, the tree editor in TEITOK is only one module amongst many, and one that is open for improvement or even replacement if better option become available. It is just not possible to use UD tools like ConlluEditor directly in TEITOK due to the different storage format (although if so desired, such tools could be used by either adopting the tool to work with XML, or using an export-import strategy).

Grew is different in two respects. On the one hand, it does much more than just tree editing. It can include non-UD information such as sound alignment. And it can create and modify dependency trees by graph (re)writing. You could use grew parse in TEITOK for parsing, but there are no methods in TEITOK for automatically changing large amounts of data with grew rewrite rules or any other rules for that matter, and purposefully so: TEITOK is intended to be usable by people with limited computational skills. And when applied accidentally, large changes can render an entire corpus useless very easily. So all structural changes in TEITOK are always made from the command line, assuming people working from the command line are more aware of what they are doing, and will for instance make a backup of their data before any global changes.

And on the other hand, Grew match provides a search directly over the storage format (CoNLL-U) without any previous indexing, while in TEITOK it is always required to index the corpus after making (large) changes, making for a sometimes cumbersome intermediate step. But (almost) all CQS use indexing, and do so to increase search speed. CWB itself natively indexes over VRT files. Tools built upon database systems, like PML-TQ using PostGres, use the native indexing of the database system (and often load the data from a different format to start with). And even tools working directly with XML files such as BlackLab[4] or corpora built in ExistDB[5] first create an index on way or another. Grew match successfully demonstrates that for existing UD treebanks, on-the-fly indexing works sufficiently fast. But UD is starting to get used in larger corpora, the largest at the moment probably being InterCorp[6], with in total over 1G tokens. And for such large corpora, on-the-fly indexing is unlikely to be fast enough. So in the long run, indexing is likely always to be a necessary step.

The stand-off tools are based on the idea that you should keep the original data unmodified, and keep all annotation in fully independent files. That means a fundamentally different type of representation than used in CoNLL-U[7]. Stand-off annotation has both advantages and disadvantages which go well beyond the purpose of this paper. But although tools based on stand-off annotation like BRAT and ANNIS can contain all the information required for dependency trees, and as such can be used to create UD treebanks, they do so in a fundamentally different manner. The fact that TEITOK uses a format that is very close to CoNLL-U makes it closer to a native UD tool than a stand-off solution can be.

3 UDWiki

UDWiki is a TEITOK meta-project, which is to say a project that defines common settings for all projects under its umbrella. It predefines all the settings needed to create a UD treebank, making all UDWiki use the same set-up to increase consistency. It also contains a collection of specific modules that are custom build to help in the process of creating, annotating, and publishing UD treebanks. The annotation process will be described in the next section. The system is currently fully operational, yet still in beta, and can be found at: https://quest.ms.mff.cuni.cz/teitok-dev/teitok/udwiki/index.php

The idea behind the workflow of UDWiki is as follows: users can ask for the creation of a new UDWiki treebank. We will then create a TEITOK project within the UDWiki meta-project, as well as a UD git repository. The user is then asked to fill in several forms about the project, with the required metadata,

[4] http://inl.github.io/BlackLab/

[5] http://exist-db.org/

[6] https://intercorp.korpus.cz/

[7] Although the more recent options to keep character ranges in UD (TokenRange) move more in this direction

and in the case of a new language also the data needed for the language description. The user can also add additional users in the case of collaborative projects.

Once set-up, the users can start to fill their corpus with documents, and start annotating them. And once there are annotated corpus texts, the whole corpus can be pushed to the UD git repository, exporting the TEI/XML files to CoNLL-U, dividing them into development, test, and train sections, and compiling the additional files such as the README, the LICENCE and the CONTRIBUTING.

UDWiki was built upon CorpusWiki, and our experience in that project is that many people are not content building their own work on the server of someone else. That is likely to be less of an issue for UDWiki since the UDWiki data are always available via the git (although there are always cases where people want to keep the data private, for instance for PhD projects, where students often do not want their data to be public until the defense). That is why we make it explicitly possible for people to set-up their corpus in UDWiki, but download the entire project at any point if they choose to rather continue the work on a local TEITOK installation. And we also intend to keep the UDWiki system itself open source so that people can run it locally if so desired, although the amount of tools used in the background will likely make it not easy to set-up a full local version of UDWiki.

As an environment for UD treebanks, UDWiki attempts to provide an easy interface to use as many of the UD tools as possible. Where possible, those tools will be rewritten to work directly with the TEI/XML format, while for other tools the TEITOK data are first exported to CoNLL-U, after which the tool is applied. A good example of a CoNLL-U based tool is the official UD validation tool. Since the validator is an official tool, and frequently updated, it is not a good idea to make any modifications to the tool. Rather, the tool is used directly from a Git clone of the official tool repository[8] over an export of the XML file to CoNLL-U. An example of the output of the validator in UDWiki is given in figure 4, taken from the tentative treebank for Papiamento (see section 4.2). The first line specifies the name of the XML file we are editing (nanzi_kriki.xml). The third line states that there is no language definition for Papiamento in UD yet - with UDWiki rather working with a local definition that should be moved to the UD repository once sufficiently established. And below that is the raw output of the validator script. To make it easy to correct the errors encountered by the script, the pointers to the .conllu file have been replaced by hyperlinks to the original XML: so the `Line 6` in the first error line links to the editor for the corresponding token, which has the ID w-8 in nanzi_kriki.xml. And the `s-1` after it links to the tree editor for the first sentence in that same file (shown in figure 5).

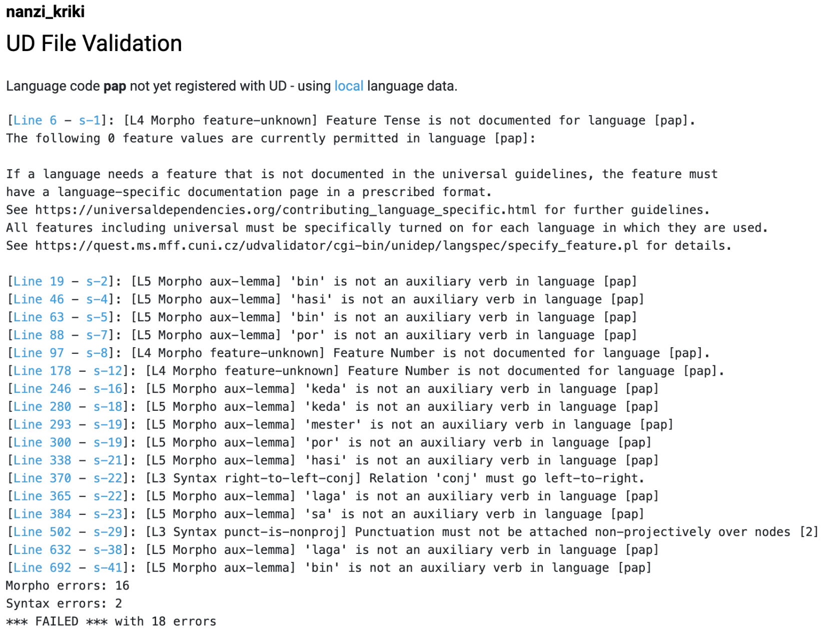
UDWiki is intended as an open-ended system. Which extensions will be added will depend on the needs of the community. For instance, we have been told that some groups working on UD treebanks would want to see their treebanks to be searchable while they are working on them, and not having to wait until the next release of UD. Mostly such teams will have their own workflow and might not want to switch to using TEITOK. For those cases, it will be possible to set-up a dummy UDWiki project that is not maintained in UDWiki, but for which the XML files are pulled from the Git repository, converted to TEI automatically and then made searchable in the various CQS.

3.1 Annotating

TEITOK has a built-in tree editor for dependency trees. The editor draws the tree, and then lets you either reattach nodes or change their dependency labels. The interface is shown in figure 5 which shows the name of the file, the sentence itself (as an XML fragment), and any sentence metadata below that, such as an English gloss in this case. By clicking on a node, you can reattach it to any other node. By clicking on a dependency label (as in the example on the *amod* label) you can select a new dependency label from a drop-down list. And by clicking on the word in the sentence, you can edit token annotations as shown in figure 2. You can also edit the metadata for the sentence which will pop up an HTML form.

For languages for which there already is a UD treebank, it is possible to run UDPIPE (Straka and Straková, 2016) to automatically parse the files in the corpus using a simple click. That will export the corpus to CoNLL-U, run it through the UDPIPE REST service, and load the results back into the TEI/XML file. Once parsed, the sentences can then optionally be manual verified. When manually

[8] `https://github.com/universaldependencies/tools`



nanzi_kriki

UD File Validation

Language code **pap** not yet registered with UD - using local language data.

```
[Line 6 - s-1]: [L4 Morpho feature-unknown] Feature Tense is not documented for language [pap].
The following 0 feature values are currently permitted in language [pap]:

If a language needs a feature that is not documented in the universal guidelines, the feature must
have a language-specific documentation page in a prescribed format.
See https://universaldependencies.org/contributing_language_specific.html for further guidelines.
All features including universal must be specifically turned on for each language in which they are used.
See https://quest.ms.mff.cuni.cz/udvalidator/cgi-bin/unidep/langspec/specify_feature.pl for details.

[Line 19 - s-2]: [L5 Morpho aux-lemma] 'bin' is not an auxiliary verb in language [pap]
[Line 46 - s-4]: [L5 Morpho aux-lemma] 'hasi' is not an auxiliary verb in language [pap]
[Line 63 - s-5]: [L5 Morpho aux-lemma] 'bin' is not an auxiliary verb in language [pap]
[Line 88 - s-7]: [L5 Morpho aux-lemma] 'por' is not an auxiliary verb in language [pap]
[Line 97 - s-8]: [L4 Morpho feature-unknown] Feature Number is not documented for language [pap].
[Line 178 - s-12]: [L4 Morpho feature-unknown] Feature Number is not documented for language [pap].
[Line 246 - s-16]: [L5 Morpho aux-lemma] 'keda' is not an auxiliary verb in language [pap]
[Line 280 - s-18]: [L5 Morpho aux-lemma] 'keda' is not an auxiliary verb in language [pap]
[Line 293 - s-19]: [L5 Morpho aux-lemma] 'mester' is not an auxiliary verb in language [pap]
[Line 300 - s-19]: [L5 Morpho aux-lemma] 'por' is not an auxiliary verb in language [pap]
[Line 338 - s-21]: [L5 Morpho aux-lemma] 'hasi' is not an auxiliary verb in language [pap]
[Line 370 - s-22]: [L3 Syntax right-to-left-conj] Relation 'conj' must go left-to-right.
[Line 365 - s-22]: [L5 Morpho aux-lemma] 'laga' is not an auxiliary verb in language [pap]
[Line 384 - s-23]: [L5 Morpho aux-lemma] 'sa' is not an auxiliary verb in language [pap]
[Line 502 - s-29]: [L3 Syntax punct-is-nonproj] Punctuation must not be attached non-projectively over nodes [2]
[Line 632 - s-38]: [L5 Morpho aux-lemma] 'laga' is not an auxiliary verb in language [pap]
[Line 692 - s-41]: [L5 Morpho aux-lemma] 'bin' is not an auxiliary verb in language [pap]
Morpho errors: 16
Syntax errors: 2
*** FAILED *** with 18 errors
```

Figure 4: The UDWiki interface for the validator

correcting trees, UDWiki keeps track of the status of each sentences, which can be explicitly set to correct or incorrect, to get an overview of the progress.

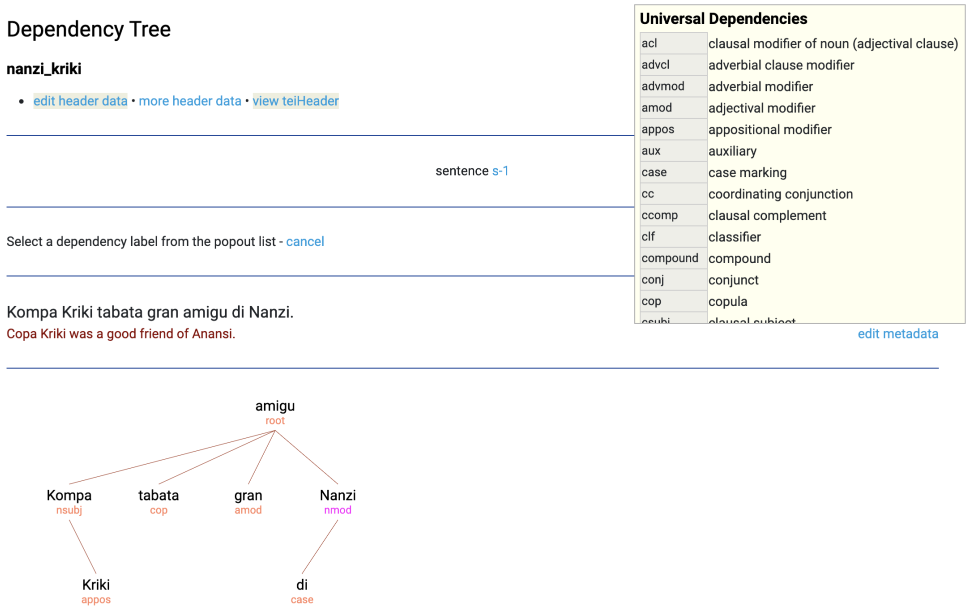
But UDWiki is explicitly set-up to help people to build treebanks for new languages. And for those there will be no models in UDPIPE (or typically any other UD parsers). For such corpora, UDWiki offers the functionally inherited from the CorpusWiki project: to use an incrementally built POS tagger and/or parser. In this way, TEITOK can be used to manually annotate corpora with UD.

3.1.1 Incremental Tagging and Parsing

The way incremental tagging in CorpusWiki worked is that the corpus is started with a single file, which is manually annotated. Since manual annotation is slow and labour-intensive, the recommendation was to start with a short file, in the range of 500-1000 tokens, ideally a translation/retelling of a popular tale. Once the initial file was fully annotated, the interface provided the option to train the NeoTag POS tagger (Janssen, 2012) on that single file, and use the resulting parameters to automatically tag the second file in the corpus. The accuracy of that tagger will naturally be low, in our experience in CorpusWiki, typically somewhere around 40% for an initial training file of 800 tokens. Yet even with low accuracy, this means that the second file is easier to annotate than the first. And by retraining the tagger after each new file that has been corrected, the accuracy rises quickly, typically reaching 90% accuracy after the first 5000 tokens.

For UDWiki, a similar set-up has been created around the UDPIPE parser. For each text in the treebank, the annotation status is kept in the header: whether it is only tokenized, POS tagged (and ideally verified), or parsed. A CoNLL-U file is created for all the tagged files, and a separate one for all the parsed files. From those files, a tagger model and a parser model are trained separately, since especially initially, there might be many more text that have been tagged than parsed. And the models can then be used to tag and parse the next file, correct, and retrain. This makes it possible to have a more accurate tagger and parser with each new file as in the Corpuswiki set-up.

The choice of UDPipe for the parsing is because it is a convenient and well-established tool that is furthermore developed within the same institute as UDWiki. But if other tools prove to be more efficient for the process, it is easy enough to modify the workflow to work with another parser. The parser just has to be trainable automatically from CoNLL-U files without manual intervention, and be sufficiently fast to allow rapid subsequent training sessions.



Dependency Tree

nanzi_kriki

- edit header data • more header data • view teiHeader

sentence s-1

Select a dependency label from the popout list - cancel

Kompa Kriki tabata gran amigu di Nanzi.

Copa Kriki was a good friend of Anansi.

edit metadata

Universal Dependencies

acl	clausal modifier of noun (adjectival clause)
advcl	adverbial clause modifier
advmod	adverbial modifier
amod	adjectival modifier
appos	appositional modifier
aux	auxiliary
case	case marking
cc	coordinating conjunction
ccomp	clausal complement
clf	classifier
compound	compound
conj	conjunct
cop	copula
csubj	clausal subject

Figure 5: The UDWiki Tree Editor

3.1.2 Guided Tagging and Parsing

Starting to build a treebank is not easy - there are a lot of things to get used to. People not used to POS tagging will have to get used to cases where it is not straightforward to tell the POS of a given example; UD has interpretations of features that do not always coincide with traditional grammar books; and the logic of the direction of a dependency relation can go contrary to the order in a constituency tree. The idea behind UDWiki is to provide as much interactive feedback to the user as possible to overcome those issues: to provide explanations about features and their values, but also to show which tags were already used for a given word in the treebank - or in other treebanks for the same language, or even in treebanks for similar languages. And not only for morphosyntax, but also for dependency relations: which is the most typical dependency relation for a given type of word, with examples. And it can fill in the same features for identical words without having to copy them to each word, which especially for the first document makes the process a lot faster.

The standard TEITOK interface can already provide many of those functions. But we are working on several interface modules specific for UD that makes this process more streamlined, and speed up the annotation process. For new languages, there is also an editor to set up the language definitions (initially in local files): to define the auxiliary verbs, the features and their values. The verification tool in figure 4 helps to pinpoint any errors in the current XML file. And when editing dependency trees, we are setting up on-the-fly verification so that the system warns immediately about any deviations from the UD norms in the current tree.

Like with the rest of the design of UDWiki, exactly which methods are most effective is something that will only become apparent by the system being used, to find out where the largest bottlenecks are, and then attempt various strategies to resolve those bottlenecks.

4 Use Cases

Given that UDWiki is not yet really released, the number of use cases is still limited to some test cases and two treebanks that are still in their initial stages. But TEITOK, and its predecessor CorpusWiki, have been used for a wide range of corpora, mostly restricted to (position-based) POS tagged corpora. The system has been very positively received, with people with little to no computational background able to build their own annotated corpora. This section will review some relevant examples.

4.1 POS Tagged Corpora

There are various LRL corpora that have been built in TEITOK, including CoDiaJe[9] (Ladino), EModSar[10] (Sardinian), and LUDVIC[11] (Caboverdean). None of these corpora are using UD, nor are they annotating dependencies. But they do serve as examples of how annotated corpora can be created for new languages in TEITOK. All three LRL corpora listed above use a locally trained NeoTag tagger, using folders to indicate which files should be used as a training corpus. For POS tagged corpora, they are all of modest size, but they are of a significant size for a treebank. EModSar is the smallest, with 6K tokens, LUDVIC is considerably larger with 100K tokens, and CoDiAJe is the largest with 800K tokens. All three are still being developed further. And with a minimal amount of initial support, all three projects have been completely independent in building and annotating the corpus, using the incremental tagging described in section 3.1.1.

Of specific interest in these corpora is CoDiAJe, since it has a feature that is difficult to deal with in plain CoNLL-U: Ladino over time has been written in various different writing systems (Latin, Hebrew, and even some Cyrillic). Therefore, CoDiAJe is a mixed corpus in which not all documents are written in the same way. Nevertheless, since all tokens are provided with a normalized (romanized) orthography, it is still a homogeneous corpus. And since search results in TEITOK are rendered as XML fragments, the results (by default) show in their original writing system.

Another example of a POS tagged corpus built in TEITOK is OLDES (Janssen et al., 2017), a corpus of Old Spanish, with around 20M tokens. OLDES was not annotated from scratch with NeoTag, but rather annotated automatically with Freeling (Carreras et al., 2004), then improved manually in TEITOK, after which a NeoTag parameter set was trained on the corpus. OLDES made use of the various types of efficient editing options provided by TEITOK to improve the automatically assigned tags.

4.2 Treebanks

The only two treebanks thus-far that are aimed at becoming full UD treebanks are the spoken Occitan corpus (UD_OCI-OCOR), and a thusfar small Papiamento corpus (UD_PAP-UFAL), both available via the UDWiki website.

OCI-OCOR is not a new corpus built in UDWiki, but rather a conversion of an existing corpus: OCOR, a corpus of Occitan oral narratives (Carruthers and Vergez-Couret, 2018). The original files of OCOR were downloaded from the Zenodo repository[12], and added to a UDWiki project with minimal modification since the files are already in the TEI format. The data were then parsed using the Talismane parameters created for the UD Occitan treebank (Miletic et al., 2020). And finally, the parsed data are being manually corrected to get a gold standard treebank for spoken Occitan. Since there already was a UD parser for Occitan, OCI-OCOR does not really show the full intent of UDWiki. But it does show that UDWiki is easy to use for languages for which there already is a parser. And it helped to provide easy access to a valuable corpus that was thus-far hard to use and not searchable online.

PAP-UFAL is a thus-far small treebank built from several texts in Papiamento collected from the web, containing at the moment two annotated texts of 1700 tokens in total. The treebank has been built from scratch in TEITOK, starting from HTML documents, and tagging and parsing them directly in the tool. It is a proof-of-concept treebank for a language for which very few (NLP) resources exist. To make the treebank accessible for non-speaker of the language, both sentences and words are glossed with English translations. The first file in the Papiamento treebank was annotated manually, and the second file was tagged automatically with the UDPipe model trained on the first text. The results are as good as one might expect.

[9] http://corptedig-glif.upf.edu/teitok/codiaje/

[10] http://corpora.unica.it/TEITOK/emodsar/index.php

[11] http://teitok.clul.ul.pt/ludvic/

[12] https://zenodo.org/record/1451753\#.YUxI1p4zblw

5 Conclusion

As shown in this paper, UDWiki is a complete environment for building UD treebanks from scratch, especially for LRL, without requiring much computational knowledge from the users. This will hopefully attract linguists interested in creating treebanks for new languages, which otherwise would not have managed to create one, or allow native speakers of languages to help out in the creation and extension of existing treebanks.

UDWiki was conceived as way to use TEITOK to generate normal UD treebanks, to be exported as CoNLL-U files and included in the UD infrastructure. But as shown, TEITOK files themselves can contain more information than CoNLL-U allows, information that has to be either removed or flattened down when exporting to CoNLL-U. So TEITOK can also be used as a richer storage format for UD treebanks, where the dependencies and morphosyntactic information is stored according to the UD standards, while all other information is stored in whichever format is most appropriate for it, whether it be metadata, spoken annotations, time mark-up, or anything else. To do that consistently, it would of course be necessary to establish annotation standards and add consistency check for everything considered core content of the extended treebanks.

An initiative like UDWiki only works if it is being used - only with hands-on experience and feedback can the system be fine-tuned to work in an optimal way. There are various questions that have not been established, such as how to use the Git repository: whether pushing the data to the repository should be done constantly, with a pre-determined frequency (say daily) or upon user request. Whether the UDWiki repositories should be kept separate from the core UD repositories (and then synced) or not. And whether only the CoNLL-U files should be exported, or also the TEITOK/XML files. And as mentioned before, the functionality of the interface is kept purposefully open in order to be able to demands coming from the community. And those needs could be of various types: changes to the interface, using a better parser, changing the workflow, adding more verification and helping tool, or adding tools for areas that are not part of UD but very helpful for LRL (or other) corpora.

An example of an area that might have to be added to UDWiki is Interlinear Glossed Texts (IGT): the experience with working with small languages has shown that without any description of their morphology, it is often impossible to start tagging immediately. The first step (field work) is to collect texts and establish the morphology, which is typically done in IGT. TEITOK has support for IGT, so in order to allow people to directly build treebanks for such languages, it might be necessary in UDWiki to let people initially create an IGT corpus, and then guide them in converting that IGT corpus to UD, not by removing the IGT data but rather by adding the UD attributes to existing IGT corpus in TEITOK.

And finally in TEITOK, we are gradually moving to make UDPipe the default tagger (and parser). There are many different types of corpora in TEITOK: historical, learner, dialect, spoken, with often additional annotation layers such error annotation, named entity tagging, rhyme schemes, etc. By having UDPipe as the default tagger, there is hence effectively a growing number of UD treebanks of different types, for document types where that would not easily happen if the treebanks would have to be written in CoNLL-U, since the additional annotations required could not (easily) be incorporated.

References

Xavier Carreras, Isaac Chao, Lluís Padró, and Muntsa Padró. 2004. Freeling: An open-source suite of language analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.

Janice Carruthers and Marianne Vergez-Couret. 2018. Méthodologie pour la constitution d'un corpus comparatif de narration orale en Occitan : objectifs, défis, solutions. *Corpus*.

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, May. European Language Resources Association (ELRA).

Kaja Dobrovoljc, Tomaž Erjavec, and Nikola Ljubešić. 2019. Improving UD processing via satellite resources for morphology. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 24–34, Paris, France, August. Association for Computational Linguistics.

Stefan Evert and Andrew Hardie. 2011. Twenty-first century corpus workbench: Updating a query architecture for the new millennium. In *Corpus Linguistics 2011.*

Bruno Guillaume. 2019. Graph Matching for Corpora Exploration. In *JLC 2019 - 10èmes Journées Internationales de la Linguistique de corpus*, Grenoble, France, November.

Johannes Heinecke. 2019. ConlluEditor: a fully graphical editor for Universal dependencies treebank files. In *Universal Dependencies Workshop 2019*, Paris.

Maarten Janssen, Josep Ausensi, and Josep Fontana. 2017. Improving POS tagging in Old Spanish using TEITOK. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pages 2–6, Gothenburg, May. Linköping University Electronic Press.

Maarten Janssen. 2012. Neotag: a pos tagger for grammatical neologism detection. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *LREC*, pages 2118–2124. European Language Resources Association (ELRA).

Maarten Janssen. 2016a. Pos tagging and less resources languages individuated features in corpuswiki. In Zygmunt Vetulani, Hans Uszkoreit, and Marek Kubis, editors, *Human Language Technology. Challenges for Computer Science and Linguistics*, pages 411–419, Cham. Springer International Publishing.

Maarten Janssen. 2016b. TEITOK: Text-faithful annotated corpora. *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, pages 4037–4043.

Maarten Janssen. 2018. TEITOK as a tool for dependency grammar. *Procesamiento del Lenguaje Natural*, 61:185–188.

Maarten Janssen. 2021. A corpus with wavesurfer and tei: Speech and video in teitok. In Kamil Ekštein, František Pártl, and Miloslav Konopík, editors, *Text, Speech, and Dialogue*, pages 261–268, Cham. Springer International Publishing.

Thomas Krause and Amir Zeldes. 2016. Annis3: A new architecture for generic corpus query and visualization. *Digital Scholarship in the Humanities*, 31(1):118–139.

Aleksandra Miletic, Myriam Bras, Marianne Vergez-Couret, Louise Esher, Clamença Poujade, and Jean Sibille. 2020. Building a Universal Dependencies treebank for Occitan. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2932–2939, Marseille, France, May. European Language Resources Association.

Petr Pajas, Jan Štěpánek, and Michal Sedlák. 2009. PML tree query. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: A web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, page 102–107.

Milan Straka and Jana Straková. 2016. UDPipe. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Minor changes make a difference: a case study on the consistency of UD-based dependency parsers

Dmytro Kalpakchi
Division of Speech, Music and Hearing
KTH Royal Institute of Technology
Stockholm, Sweden
dmytroka@kth.se

Johan Boye
Division of Speech, Music and Hearing
KTH Royal Institute of Technology
Stockholm, Sweden
jboye@kth.se

Abstract

Many downstream applications are using dependency trees, and are thus relying on dependency parsers producing correct, or at least consistent, output. However, dependency parsers are trained using machine learning, and are therefore susceptible to unwanted inconsistencies due to biases in the training data. This paper explores the effects of such biases in four languages – English, Swedish, Russian, and Ukrainian – though an experiment where we study the effect of replacing numerals in sentences. We show that such seemingly insignificant changes in the input can cause large differences in the output, and suggest that data augmentation can remedy the problems.

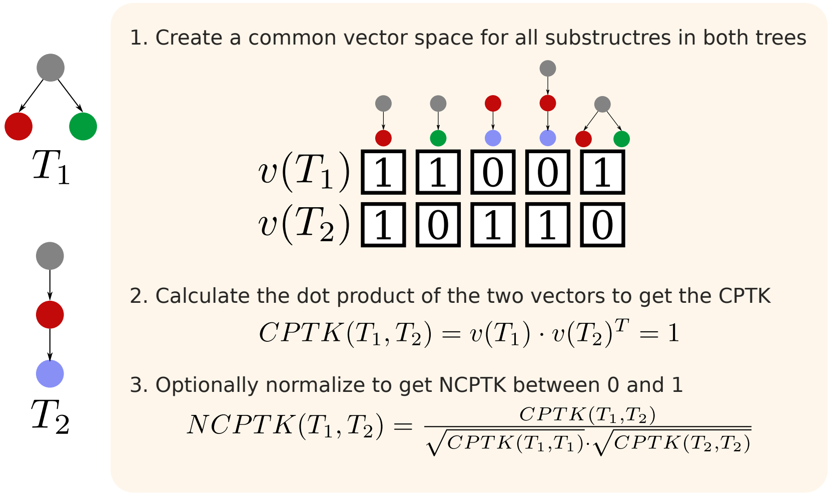
1 Introduction

The Universal Dependencies (UD) resources have steadily grown over the years, and now treebanks for over 100 languages are available. The UD community has made a tremendous effort in providing a rich toolset for utilizing the treebanks for downstream applications, including pre-trained models for dependency parsing (Straka et al., 2016; Qi et al., 2020) and tools for manipulating UD trees (Popel et al., 2017; Peng and Zeldes, 2018; Kalpakchi and Boye, 2020).

Such an extensive infrastructure makes it more appealing to develop multilingual downstream applications based on UD, as a deterministic and more explainable competitor to the currently dominant neural methods. It is also compelling to use UD-based metrics for evaluation in multilingual settings. In fact, researchers have already started exploring such possibilities on both mentioned tracks. Kalpakchi and Boye (2021) proposed a UD-based multilingual method for generating reading comprehension questions. Chaudhary et al. (2020) designed a UD-based method for automatically extracting rules governing morphological agreement. Pratapa et al. (2021) proposed a UD-based metric to evaluate the morphosyntactic well-formedness of generated texts.

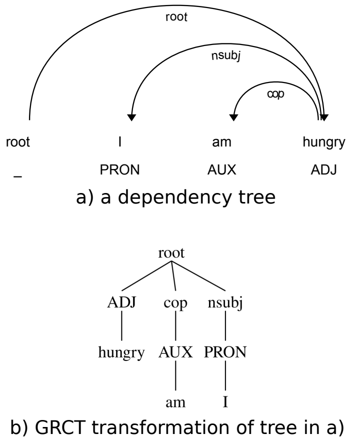
The authors of the latter two articles trained their own more robust versions of the dependency parsers, suitable for their needs. The authors of the first article relied on the off-the-shelf model, making the robustness of pre-trained dependency parsers crucial for the success of the downstream applications. For instance, sentence simplification rules based on dependency trees might simply not fire due to a mistakenly identified head or dependency relation. In fact, state-of-the-art dependency parsers are somewhat error-prone and not perfect, and assuming otherwise might potentially harm the performance of downstream applications. A more relaxed (and realistic) assumption is that the errors made by the parser are at least *consistent*, so that potentially useful patterns for the task at hand can still be inferred from data. These patterns might not always be linguistically motivated, but if the dependency parser makes consistent errors, they can still be useful for the task at hand.

In this article, we perform a case study operating under this relaxed assumption and investigate the consistency of errors while parsing sentences containing numerals. This step is useful, for instance, in question generation (especially for reading comprehension in the history domain) or numerical entity identification (e.g., distinguishing years from weights or distances).



T_1
T_2
1. Create a common vector space for all substructres in both trees
$v(T_1)$ 1 1 0 0 1
$v(T_2)$ 1 0 1 1 0
2. Calculate the dot product of the two vectors to get the CPTK
$CPTK(T_1, T_2) = v(T_1) \cdot v(T_2)^T = 1$
3. Optionally normalize to get NCPTK between 0 and 1
$NCPTK(T_1, T_2) = \frac{CPTK(T_1,T_2)}{\sqrt{CPTK(T_1,T_1)} \cdot \sqrt{CPTK(T_2,T_2)}}$

Figure 1: A simple example illustrating *the concept* behind convolution partial tree kernels (in practice the vector space is induced only implicitly and CPTK is calculated using dynamic programming)



a) a dependency tree

b) GRCT transformation of tree in a)

Figure 2: A simple example of a GRCT transformation

2 Background: Convolution partial tree kernels

In order to measure parser accuracy, metrics like Unlabelled or Labelled Attachment Score (UAS and LAS, respectively) are often used. However, these metrics they do not fully reflect the usefulness of the parsers in downstream applications. A minor error in attaching one dependency arc will result in a minor decrease in UAS and LAS. In fact, the very same minor error might lead to a completely unusable tree for the task at hand, depending on how close the error is to the root. Therefore, we need a metric that penalizes errors more the closer the errors are to the root.

One metric possessing this desirable property is the convolution partial tree kernel (CPTK), originally proposed by Moschitti (2006) as a similarity measure for dependency trees. The basic idea is to represent trees as vectors in a common vector space, in such a way that the more common substructures two given trees have, the higher the dot product is between the corresponding two vectors (as illustrated in Figure 1). However, the vector space is induced only implicitly, whereas the dot product (the CPTK) itself is calculated using a dynamic programming algorithm (for more details we refer to the original article). CPTK values increase with the size of the trees, and thus can take any non-negative values, making them hard to interpret. Hence, we use normalized CPTK (NCPTK) which takes values between 0 and 1, and is calculated as shown in Figure 1.

However, CPTKs can not handle labeled edges and were originally applied to dependency trees containing only lexicals. In this article, we use an extension proposed by Croce et al. (2011), which includes edge labels (DEPREL) as separate nodes. The resulting computational structure, the Grammatical Relation Centered Tree (GRCT), is illustrated in Figure 2. A dependency tree is transformed into a GRCT by making each UPOS node a child of a DEPREL node and a father of a FORM node.

3 Method

To explore the consistency of errors while parsing numerals, we have used UD treebanks for 4 European languages (2 Germanic and 2 Slavic). To simplify, we considered only sentences containing numerals representing years, later referred to as *original sentences*. We defined these numerals as 4 digits surrounded by spaces, via the simple regular expression `"(?<= )\d{4}(?= )"`. We then sampled uniformly at random 50 integers between 1100 and 2100 using a fixed random seed, and replaced the occurrences of the previously identified numerals in the original sentences by each of these numbers. Thus, for every found original sentence in a treebank, we synthesized 50 *augmented sentences* (later referred to as *an augmented batch*), only differing in the 4-digit numbers. We only substituted the first found occurrence of a 4-digit number in a sentence. However, if the same number appeared multiple times in the sentence, then all its occurrences were substituted.

Given such minor changes, a consistent dependency parser should output the same dependency tree for every sentence in each augmented batch. These trees should not necessarily be the same as gold original trees (although this is obviously desirable), but at the very least, the errors made in each augmented batch should be of the same kind. We consider two trees to have the errors of the same kind, and thus belonging to the same *cluster of errors*, if their dependency trees only differ in the 4-digit numerals. All DEPRELs, UPOS tags and FEATS should be exactly the same for any two trees in the same cluster.

Evidently, not all 4-digit numbers in the original sentences were actually years, but the argument about the consistency of errors still stands even if the numbers were amounts of money, temperatures, etc. The magnitude of the numbers was not drastically changed (they are still 4-digit numbers), so the sentences should remain intelligible also after substitution.

In order to evaluate both the consistency of errors and correctness of a dependency parser after introducing the changes above, we need to answer the following questions.

Q1 How many augmented batches are parsed completely correctly?

- if the corresponding original sentence is parsed correctly
- if the corresponding original sentence is parsed incorrectly

Q2 How many sentences in each augmented batch are parsed correctly on average?

- if the corresponding original sentence is parsed correctly
- if the corresponding original sentence is parsed incorrectly

Q3 How many augmented batches corresponding to incorrectly parsed original sentences have consistent errors, i.e. have the same dependency trees within a batch except FORMs and LEMMAs?

Q4 On average, how many clusters of errors does an augmented batch with inconsistent errors have?

Q5 On average, how similar are dependency trees in the clusters found in Q4?

Answering Q1 to Q3 is trivial by parsing original and augmented sentences using a pre-trained dependency parser and calculating descriptive statistics. To answer Q4 and Q5, we propose to calculate NCPTK for each pair of trees in an augmented batch. To perform the calculations, we transform each dependency tree to GRCT replacing FORMs (which will be different by experimental design) with the FEATS. We can then construct an undirected graph, where each node is a dependency tree in the batch and two nodes are connected if their NCPTK is exactly 1 (i.e., their dependency trees are identical). Then the problem of finding error clusters in Q4 boils down to finding all maximal cliques in the induced undirected graph, for which we use Bron–Kerbosch algorithm (Bron and Kerbosch, 1973). Similarity of dependency trees in the given clusters can be assessed using the already calculated NCPTKs, which will provide the answer to Q5.

In hopes of improving parsers' performance and consistency of errors we have also tried to retrain the tokenizer, lemmatizer, PoS tagger and dependency parser (later referred to as a *pipeline*) from scratch using two approaches. The first approach relies on *numeral augmentation* and starts by sampling 20 four-digit integers using a different random seed (while ensuring no overlap with the previously used 50 integers). Using these 20 new numbers and the same procedure as before, we synthesized 20 additional sentences per each previously found original sentence in the training and development treebanks. We will refer to treebanks formed by original and newly synthesized sentences as *augmented treebanks*. The second approach uses *token substitution* and replaces previously found four-digit integers with a special token `NNNN`. The training and development treebanks after this procedure keep their size the same (in constrast to the numeral augmentation method) and will be later referred to as *substituted treebanks*.

We have used Stanza (Qi et al., 2020) to get pretrained dependency parsers as well as to train the whole pipeline from scratch and UDon2 (Kalpakchi and Boye, 2020) to perform the necessary manipulations on dependency trees and calculate NCPTK. The code is available at `https://github.com/dkalpakchi/ud_parser_consistency`.

4 Experimental results

4.1 Pretrained pipeline

We have started the experiment by parsing all original and augmented sentences in the training and development treebanks of the respective languages. The results summary for the off-the-shelf parser are presented in Table 1. To our surprise, some sentences were not segmented correctly, i.e. one sentence became multiple, both among original and augmented sentences. However, we did not find any consistent pattern: for instance, the Swedish parser made more segmentation errors for augmented sentences, whereas all the other parsers exhibited the opposite. Nonetheless, we have excluded the cases with wrong sentence segmentation from further analysis. The final number of sentences considered is shown in the rows "Original considered" and "Augmented considered" in Table 1.

Metric	**English**		**Swedish**		**Russian**		**Ukrainian**	
	Train	Dev	Train	Dev	Train	Dev	Train	Dev
Original in total	235	14	108	5	1420	270	103	29
Wrong sent. segm.	12	0	2	0	25	5	1	1
Original considered	223	14	106	5	1395	265	102	28
Corr. parsed sent.	53	1	76	1	360	53	27	2
Corr. parsed sent. (%)	23.8%	7.1%	71.7%	20%	25.8%	20%	26.5%	7.1%
Augmented in total	11150	700	5300	250	69750	13250	5100	1400
Wrong sent. segm.	0	0	17	14	13	0	0	0
Augmented considered	11150	700	5283	236	69737	13250	5100	1400
Corr. parsed sent.	2689	50	3525	43	17787	2540	1227	100
Corr. parsed sent. (%)	24.1%	7.1%	66.7%	18.2%	25.5%	19.2%	24.1%	7.1%

Table 1: Results of parsing the original and augmented sentences with pre-trained parsers from Stanza. "Corr" stands for "Correctly", "sent" stands for sentence(s)

We have excluded metrics commonly used within UD community, e.g. UAS, LAS or BLEX, because for these metrics we observed only minor changes (less than 1 percentage point). Another argument for omitting these metrics is that while they are useful in comparing different parsers, they do not fully reflect the usefulness of the parsers in downstream applications. In fact, even a minor error in attaching one dependency arc might lead to a completely wrong tree for the task at hand (depending on how close the error is to the root). Keeping this in mind, we compared accuracy on the sentence level only (reported in the rows "Correctly parsed" in Table 1). We deemed a sentence to be correctly parsed if the NCPTK between its dependency tree and its gold counterpart was 1. We transformed all trees to GRCT and replaced FORM with FEATS, thus requiring not only all DEPREL to be identical, but also all UPOS and FEATS. As can be seen, the number of correctly parsed sentences is either on par or worse for augmented sentences, reaching a performance drop of 5 percentage points for the Swedish training set!

Results of a more detailed analysis needed for answering questions 1 - 5 (posed in Section 3) are reported in Tables 2 - 5. We adopt the following notation for these tables: "Original +" ("Original -") indicates cases when the original sentence was correctly (incorrectly) parsed. "QX" indicates a row with data necessary for answering question X, "Corr" stands for "Correct(ly)", "sent" stands for sentences.

We observe a number of interesting patterns from these reports. If the original sentences are incorrectly parsed, the vast majority of sentences in the corresponding augmented batches will also be incorrectly parsed (see mean and median in Q2 rows for "Original -"). The fact that an original sentence is correctly parsed does not mean that all sentences in augmented batches will be correctly parsed (see mean and median in Q2 rows for "Original +"). In fact, the number of wrong batches in such a case can be surprisingly large, e.g. 24 (31.5%) for the Swedish training set.

Metric	**Training set**		**Development set**	
	Original +	Original -	Original +	Original -
Batches considered	53	170	1	13
Completely corr. batches (Q1)	49	0	1	0
Corr. parsed sent. within a batch (Q2)				
Mean (SD)	49 (6.14)	0.54 (3.67)	50 (0)	0 (0)
Median (Min - Max)	50 (5 - 50)	0 (0 - 37)	50 (50 - 50)	0 (0 - 0)
Batches with consistent errors (Q3)	0	101	NA	4
Number of error clusters (Q4)				
Mean (SD)	2 (0)	2.63 (0.95)	NA	3.89 (2.64)
Median (Min - Max)	2 (2 - 2)	2 (2 - 7)	NA	3 (2 - 10)
Between-cluster NCPTK (Q5)				
Mean (SD)	0 (0)	0.07 (0.15)	NA	0.04 (0.09)
Median (Min - Max)	0 (0 - 0)	0 (0 - 0.8)	NA	0 (0 - 0.28)

Table 2: A detailed analysis of the parsing results for English using a pretrained pipeline

Metric	**Training set**		**Development set**	
	Original +	Original -	Original +	Original -
Batches considered	76	30	1	4
Completely corr. batches (Q1)	52	0	0	0
Corr. parsed sent. within a batch (Q2)				
Mean (SD)	45.05 (10.77)	3.37 (10.5)	43 (0)	0 (0)
Median (Min - Max)	50 (0 - 50)	0 (0 - 42)	43 (43 - 43)	0 (0 - 0)
Batches with consistent errors (Q3)	0	16	0	1
Number of error clusters (Q4)				
Mean (SD)	2.29 (0.68)	2.43 (1.05)	2 (0)	2.33 (0.47)
Median (Min - Max)	2 (2 - 4)	2 (2 - 5)	2 (2 - 2)	2 (2 - 3)
Between-cluster NCPTK (Q5)				
Mean (SD)	0.04 (0.12)	0.04 (0.11)	0 (0)	0.0002 (0.0003)
Median (Min - Max)	0 (0 - 0.67)	0 (0 - 0.37)	0 (0 - 0)	0 (0 - 0.0008)

Table 3: A detailed analysis of the parsing results for Swedish using a pretrained pipeline

Metric	**Training set**		**Development set**	
	Original +	Original -	Original +	Original -
Batches considered	360	1035	53	212
Completely corr. batches (Q1)	341	0	48	0
Corr. parsed sent. within a batch (Q2)				
Mean (SD)	48.85 (6.34)	0.19 (2.11)	47.87 (7.81)	0.01 (0.21)
Median (Min - Max)	50 (2 - 50)	0 (0 - 41)	50 (3 - 50)	0 (0 - 3)
Batches with consistent errors (Q3)	0	860	0	173
Number of error clusters (Q4)				
Mean (SD)	2.21 (0.69)	2.16 (0.43)	2.2 (0.4)	2.13 (0.4)
Median (Min - Max)	2 (2 - 5)	2 (2 - 4)	2 (2 - 3)	2 (2 - 4)
Between-cluster NCPTK (Q5)				
Mean (SD)	0.08 (0.18)	0.04 (0.14)	0 (0)	0.08 (0.2)
Median (Min - Max)	0 (0 - 0.67)	0 (0 - 0.75)	0 (0 - 0)	0 (0 - 0.72)

Table 4: A detailed analysis of the parsing results for Russian using a pretrained pipeline

Metric	Training set		Development set	
	Original +	Original -	Original +	Original -
Batches considered	27	75	2	26
Completely corr. batches (Q1)	24	0	2	0
Corr. parsed sent. within a batch (Q2)				
Mean (SD)	45.41 (13.14)	0.01 (0.11)	50 (0)	0 (0)
Median (Min - Max)	50 (4 - 50)	0 (0 - 1)	50 (50 - 50)	0 (0 - 0)
Batches with consistent errors (Q3)	0	52	NA	11
Number of error clusters (Q4)				
Mean (SD)	2 (0)	2.61 (1.37)	NA	2.8 (0.9)
Median (Min - Max)	2 (2 - 2)	2 (2 - 8)	NA	3 (2 - 5)
Between-cluster NCPTK (Q5)				
Mean (SD)	0 (0)	0.12 (0.22)	NA	0.06 (0.19)
Median (Min - Max)	0 (0 - 0)	0 (0 - 0.775)	NA	0 (0 - 0.77)

Table 5: A detailed analysis of the parsing results for Ukrainian using a pretrained pipeline

The errors in augmented batches are not consistent. The degree of inconsistency varies between the languages ranging from around 17% (175 of 1035) for the Russian training set to 75% (3 of 4) for the Swedish development set (see Q3 rows). The average observed inconsistency of errors is around 44%. The degree of inconsistency has a similar magnitude between the training and development sets. The most typical number of error clusters is 2 and maximum observed is 10 (see Q4 rows). The trees between the error clusters have mostly low NCPTK (see Q5 rows) indicating either a large number of errors or errors occurring early on (close to the root). We provide some examples of batches with inconsistent errors in the Appendix.

4.2 Pipeline trained from scratch on treebanks with numeral augmentation

We have repeated the same experiment as in the previous section, but with a pipeline trained from scratch on augmented treebanks (as outlined in Section 3). The results summary is reported in Table 6.

Metric	English		Swedish		Russian		Ukrainian	
	Train	Dev	Train	Dev	Train	Dev	Train	Dev
Original in total	235	14	108	5	1420	270	103	29
Wrong sent. segm.	5	0	3	0	18	5	0	0
Original considered	230	14	105	5	1402	265	103	29
Corr. parsed sent.	230	0	97	2	976	48	102	3
Corr. parsed sent. (%)	**100%**	0%	**92.4%**	**40%**	**69.6%**	18.1%	**99%**	**10.3%**
Augmented in total	11500	700	5250	250	70100	13250	5150	1450
Wrong sent. segm.	0	0	0	0	13	0	0	0
Augmented considered	11500	700	5250	250	70087	13250	5150	1450
Corr. parsed sent.	11452	0	4864	100	49005	2437	5100	133
Corr. parsed sent. (%)	**99.6%**	0%	**92.7%**	**40%**	**69.9%**	18.4%	**99%**	**9.2%**

Table 6: Results of parsing the original and augmented sentences with the pipeline trained on augmented treebanks. “Corr” stands for “Correctly”, “sent” stands for sentence(s). Performance improvements with respect to the pre-trained parser (see Table 1) are indicated in **bold**.

Retraining with numeral augmentation resulted in a clear and substantial performance boost for all languages, especially for the training treebanks. Performance boost on the development treebanks is less pronounced and sometimes leads to a slight performance degradation. We attribute this to a possible overfitting, indicating that 20 samples per an original sentence might have been too many and the procedure needs to be refined in future. Nevertheless, the detailed analysis, reported in Appendix, shows that the number of wrong sentence segmentations decreased for all languages and a consistency of errors is either better or on par with the pretrained counterparts. The number of error clusters got reduced to a maximum of 4 compared to 10 for the off-the-shelf parser.

4.3 Pipeline trained from scratch on treebanks with token substitution

We have repeated the same experiment as in the previous section, but with a pipeline trained from scratch on substituted treebanks (as outlined in Section 3). The results summary is reported in Table 7.

Metric	**English**		**Swedish**		**Russian**		**Ukrainian**	
	Train	Dev	Train	Dev	Train	Dev	Train	Dev
Substituted in total	235	14	108	5	1420	270	103	29
Wrong sent. segm.	14	0	1	0	10	1	2	1
Substituted considered	221	14	107	5	1410	269	101	28
Corr. parsed sent.	81	1	73	2	341	59	23	2
Corr. parsed sent. (%)	**36.7%**	7.1%	68.2%	**40%**	24.2%	**21.9%**	22.8%	7.1%

Table 7: Results of parsing the substituted sentences with the pipeline trained on treebanks with token susbtitution. “Corr” stands for “Correctly”, “sent” stands for sentence(s). Performance improvements with respect to the pre-trained parser (see Table 1) are indicated in **bold**.

Retraining with token substitution resulted in a slight performance boost for Russian and Swedish on the development treebanks and a slight performance degradation on the training treebanks for all languages except English. Interestingly, more sentences have been segmented correctly for Russian and Swedish, while the parsers for English and Ukrainian produce more segmentation errors compared to pre-trained parsers. At the same time, more sentences have been segmented incorrectly compared to the numeral augmentation method (except for Russian). Given that all models were re-trained with the same default seed from Stanza, we are unsure what this can be attributed to, other than the choice of the token `NNNN` itself. The tokenization model in Stanza is based on unit (character) embeddings, so a tokenization model might benefit from a token without letters or just from replacing all 4-digit numerals with one fixed integer, say 0000. This is, however, highly speculative and requires further investigation.

An obvious advantage of token substitution is that the errors become consistent (since no clusters of errors could potentially be formed). However, the observed effect on performance suggests that token substitution with this specific token `NNNN` is not the best solution to the problem.

5 Conclusion

We have observed that such a minor change as changing one 4-digit number for another leads to surprising performance fluctuations for pretrained parsers. Furthermore, we have noted the errors to be inconsistent, making the development of downstream applications more complicated. To alleviate the issue we tried out two methods and trained two proof-of-concept pipelines from scratch. One of the methods, namely the numeral augmentation scheme, resulted in substantial performance gains.

Finally, the results of the experiment suggest that UD treebanks might be biased towards specific time intervals, e.g. the 19th and 20th centuries. Bias in the data leads to bias in the models making it harder to use the parser for some downstream applications, e.g. in the history domain. The results of this experiment also prompt a further and more extensive investigation of possible other biases, such as names of geographical entities, gender pronouns, currencies, etc.

Acknowledgements

This work was supported by Vinnova (Sweden’s Innovation Agency) within the project 2019-02997. We would like to thank the anonymous reviewers for their comments and the suggestion to try token substitution.

References

Coen Bron and Joep Kerbosch. 1973. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577.

Aditi Chaudhary, Antonios Anastasopoulos, Adithya Pratapa, David R. Mortensen, Zaid Sheikh, Yulia Tsvetkov, and Graham Neubig. 2020. Automatic extraction of rules governing morphological agreement. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5212–5236, Online, November. Association for Computational Linguistics.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1034–1046, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Dmytro Kalpakchi and Johan Boye. 2020. UDon2: a library for manipulating Universal Dependencies trees. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 120–125, Barcelona, Spain (Online), December. Association for Computational Linguistics.

Dmytro Kalpakchi and Johan Boye. 2021. Quinductor: a multilingual data-driven method for generating reading-comprehension questions using universal dependencies. *arXiv preprint arXiv:2103.10121.*

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *European Conference on Machine Learning*, pages 318–329. Springer.

Siyao Peng and Amir Zeldes. 2018. All roads lead to UD: Converting Stanford and Penn parses to English Universal Dependencies with multilayer annotations. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 167–177, Santa Fe, New Mexico, USA, August. 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, May. Association for Computational Linguistics.

Adithya Pratapa, Antonios Anastasopoulos, Shruti Rijhwani, Aditi Chaudhary, David R Mortensen, Graham Neubig, and Yulia Tsvetkov. 2021. Evaluating the morphosyntactic well-formedness of generated texts. *arXiv preprint arXiv:2103.16590.*

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*, pages 101–108, Online, July. Association for Computational Linguistics.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Appendix A Details of the experimental setup

We have experimented with the training and development sets of the following treebanks: UD_English-EWT, UD_Swedish-Talbanken, UD_Russian-SynTagRus, UD_Ukrainian-IU. For sampling 50 integers used for validating the parser's performance, we have seeded Numpy's random number generator with the 1000th prime number (7919). For sampling 20 integers used for augmenting treebanks for re-training, we chosen the 999th prime number (7907) as the random seed. Then we sampled 100 integers, filtered out all overlapping with the previously sampled 50 and then taken the first 20 integers of the remainder.

Appendix B Detailed results for the pipeline trained from scratch

Metric	Training set		Development set	
	Original +	Original -	Original +	Original -
Batches considered	230	0	0	14
Completely corr. batches (Q1)	229	NA	NA	0
Corr. parsed sent. within a batch (Q2)				
Mean (SD)	49.79 (3.16)	NA	NA	0 (0)
Median (Min - Max)	50 (2 - 50)	NA	NA	0 (0 - 0)
Batches with consistent errors (Q3)	0	NA	NA	4
Number of error clusters (Q4)				
Mean (SD)	2 (0)	NA	NA	2.6 (0.8)
Median (Min - Max)	2 (2 - 2)	NA	NA	2 (2 - 4)
Between-cluster NCPTK (Q5)				
Mean (SD)	0 (0)	NA	NA	0.05 (0.1)
Median (Min - Max)	0 (0 - 0)	NA	NA	0 (0 - 0.31)

Table 8: A detailed analysis of the parsing results for English using a retrained pipeline

Metric	Training set		Development set	
	Original +	Original -	Original +	Original -
Batches considered	97	8	2	3
Completely corr. batches (Q1)	97	0	2	0
Corr. parsed sent. within a batch (Q2)				
Mean (SD)	50 (0)	1.75 (4.63)	50 (0)	0 (0)
Median (Min - Max)	50 (50 - 50)	0 (0 - 14)	50 (50 - 50)	0 (0 - 0)
Batches with consistent errors (Q3)	NA	7	NA	1
Number of error clusters (Q4)				
Mean (SD)	NA	3 (0)	NA	2 (0)
Median (Min - Max)	NA	3 (3 - 3)	NA	2 (2 - 2)
Between-cluster NCPTK (Q5)				
Mean (SD)	NA	0 (0)	NA	0.04 (0.04)
Median (Min - Max)	NA	0 (0 - 0)	NA	0.04 (0 - 0.08)

Table 9: A detailed analysis of the parsing results for Swedish using a retrained pipeline

Metric	Training set		Development set	
	Original +	Original -	Original +	Original -
Batches considered	976	426	48	217
Completely corr. batches (Q1)	950	1	44	0
Corr. parsed sent. within a batch (Q2)				
Mean (SD)	49.58 (3.63)	1.44 (7.75)	49.77 (0.92)	0.22 (2.92)
Median (Min - Max)	50 (2 - 50)	0 (0 - 50)	50 (45 - 50)	0 (0 - 43)
Batches with consistent errors (Q3)	0	369	0	149
Number of error clusters (Q4)				
Mean (SD)	2.08 (0.27)	2.09 (0.34)	2 (0)	2.13 (0.4)
Median (Min - Max)	2 (2 - 3)	2 (2 - 4)	2 (2 - 2)	2 (2 - 4)
Between-cluster NCPTK (Q5)				
Mean (SD)	0.05 (0.14)	0.08 (0.18)	0.13 (0.22)	0.07 (0.2)
Median (Min - Max)	0 (0 - 0.5)	0 (0 - 0.67)	0.003 (0 - 0.5)	0 (0 - 0.87)

Table 10: A detailed analysis of the parsing results for Russian using a retrained pipeline

Metric	Training set		Development set	
	Original +	Original -	Original +	Original -
Batches considered	102	1	3	26
Completely corr. batches (Q1)	102	0	2	0
Corr. parsed sent. within a batch (Q2)				
Mean (SD)	50 (0)	0 (0)	44.33 (8.01)	0 (0)
Median (Min - Max)	50 (50 - 50)	0 (0 - 0)	50 (33 - 50)	0 (0 - 0)
Batches with consistent errors (Q3)	NA	1	0	13
Number of error clusters (Q4)				
Mean (SD)	NA	NA	2 (0)	2.46 (0.75)
Median (Min - Max)	NA	NA	2 (2 - 2)	2 (2 - 4)
Between-cluster NCPTK (Q5)				
Mean (SD)	NA	NA	0.29 (0)	0.09 (0.22)
Median (Min - Max)	NA	NA	0.29 (0.29 - 0.29)	0 (0 - 0.67)

Table 11: A detailed analysis of the parsing results for Ukrainian using a retrained pipeline

Appendix C Examples of batches with inconsistent errors

In this section we report dependency trees from the augmented batch with the largest observed number of error clusters (which happened to be 10 clusters for the English development set). The original sentences in these clusters were too long, so we have pruned the dependency trees to include only the differing subtrees. The cluster sizes and included numerals are as follows:

Cluster 1. 2 trees (numerals 1505, 1505)

Cluster 2. 3 trees (numerals 1798, 1777, 1817)

Cluster 3. 3 trees (numerals 1872, 1844, 1883)

Cluster 4. 3 trees (numerals 1361, 1338, 1427)

Cluster 5. 4 trees (numerals 1704, 1605, 1662, 1562)

Cluster 6. 5 trees (numerals 1420, 1344, 1295, 1504, 1299)

Cluster 7. 5 trees (numerals 1625, 1599, 1564, 1564, 1493)

Cluster 8. 6 trees (numerals 1128, 2024, 1147, 1182, 2030, 1205)

Cluster 9. 7 trees (numerals 1964, 1308, 1415, 1413, 1404, 1967, 1413)

Cluster 10. 8 trees (numerals 1774, 1721, 1759, 1759, 1461, 1731, 1724, 1832)

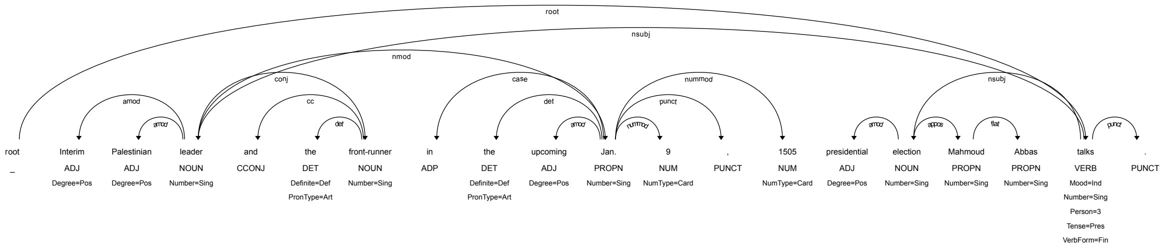


Figure 3: An example truncated dependency tree from cluster 1

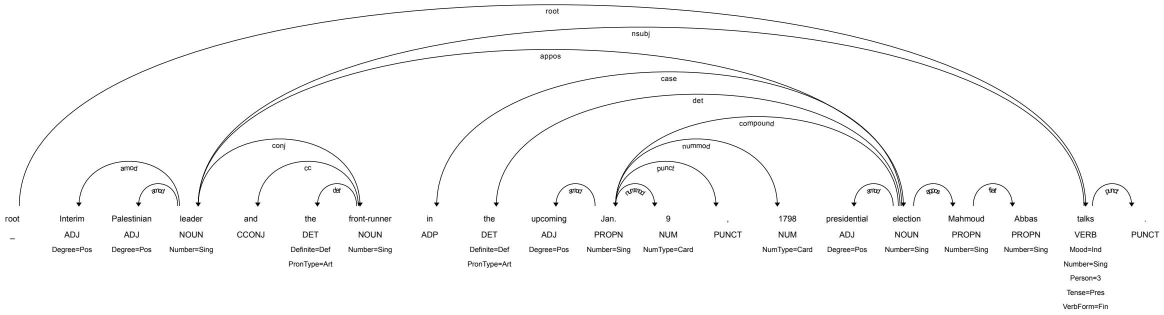

root
nsubj
appos
case
det
compound
nummod
conj
punct
amod
cc
amod
det
amod
nummod
amod
appos
flat
punct
root Interim Palestinian leader and the front-runner in the upcoming Jan. 9 , 1798 presidential election Mahmoud Abbas talks .
_ ADJ ADJ NOUN CCONJ DET NOUN ADP DET ADJ PROPN NUM PUNCT NUM ADJ NOUN PROPN PROPN VERB PUNCT
Degree=Pos Degree=Pos Number=Sing Definite=Def Number=Sing Definite=Def Degree=Pos Number=Sing NumType=Card NumType=Card Degree=Pos Number=Sing Number=Sing Number=Sing Mood=Ind
PronType=Art PronType=Art Number=Sing
Person=3
Tense=Pres
VerbForm=Fin

Figure 4: An example truncated dependency tree from cluster 2

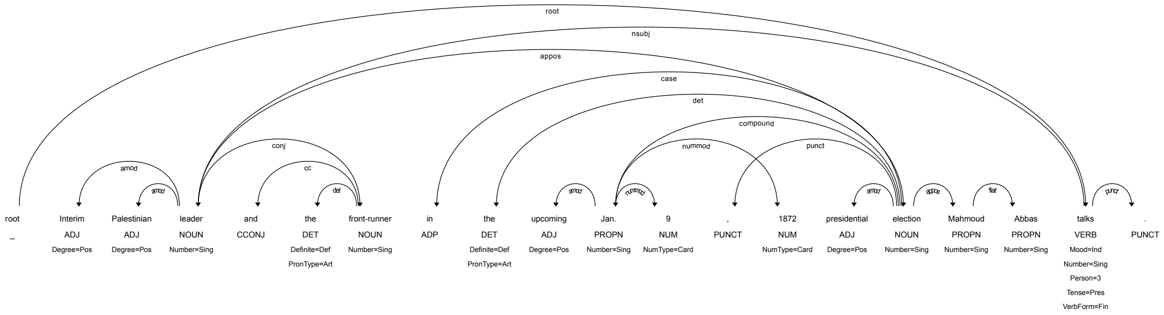

root
nsubj
appos
case
det
compound
nummod
punct
conj
cc
amod
amod
det
amod
nummod
amod
appos
flat
punct
root Interim Palestinian leader and the front-runner in the upcoming Jan. 9 , 1872 presidential election Mahmoud Abbas talks .
_ ADJ ADJ NOUN CCONJ DET NOUN ADP DET ADJ PROPN NUM PUNCT NUM ADJ NOUN PROPN PROPN VERB PUNCT
Degree=Pos Degree=Pos Number=Sing Definite=Def Number=Sing Definite=Def Degree=Pos Number=Sing NumType=Card NumType=Card Degree=Pos Number=Sing Number=Sing Number=Sing Mood=Ind
PronType=Art PronType=Art Number=Sing
Person=3
Tense=Pres
VerbForm=Fin

Figure 5: An example truncated dependency tree from cluster 3

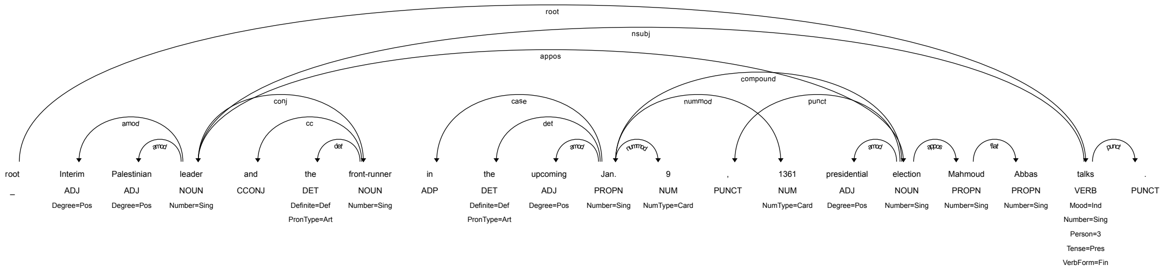

root
nsubj
appos
compound
conj
case
nummod
punct
amod
cc
det
amod
det
amod
nummod
amod
appos
flat
punct
root Interim Palestinian leader and the front-runner in the upcoming Jan. 9 , 1361 presidential election Mahmoud Abbas talks .
_ ADJ ADJ NOUN CCONJ DET NOUN ADP DET ADJ PROPN NUM PUNCT NUM ADJ NOUN PROPN PROPN VERB PUNCT
Degree=Pos Degree=Pos Number=Sing Definite=Def Number=Sing Definite=Def Degree=Pos Number=Sing NumType=Card NumType=Card Degree=Pos Number=Sing Number=Sing Number=Sing Mood=Ind
PronType=Art PronType=Art Number=Sing
Person=3
Tense=Pres
VerbForm=Fin

Figure 6: An example truncated dependency tree from cluster 4

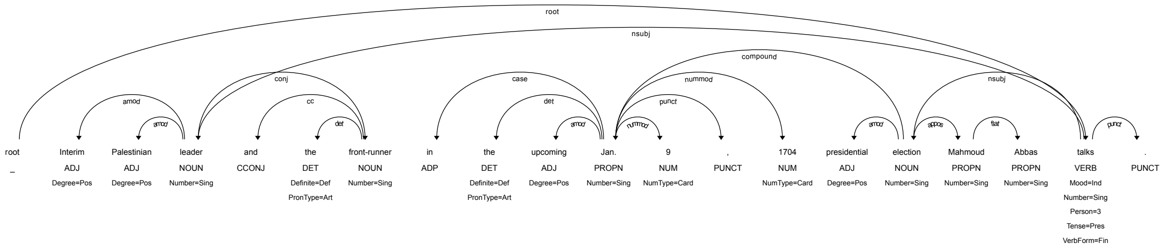

root
nsubj
compound
conj
case
nummod
nsubj
amod
cc
det
punct
amod
det
amod
nummod
amod
appos
flat
punct
root Interim Palestinian leader and the front-runner in the upcoming Jan. 9 , 1704 presidential election Mahmoud Abbas talks .
_ ADJ ADJ NOUN CCONJ DET NOUN ADP DET ADJ PROPN NUM PUNCT NUM ADJ NOUN PROPN PROPN VERB PUNCT
Degree=Pos Degree=Pos Number=Sing Definite=Def Number=Sing Definite=Def Degree=Pos Number=Sing NumType=Card NumType=Card Degree=Pos Number=Sing Number=Sing Number=Sing Mood=Ind
PronType=Art PronType=Art Number=Sing
Person=3
Tense=Pres
VerbForm=Fin

Figure 7: An example truncated dependency tree from cluster 5

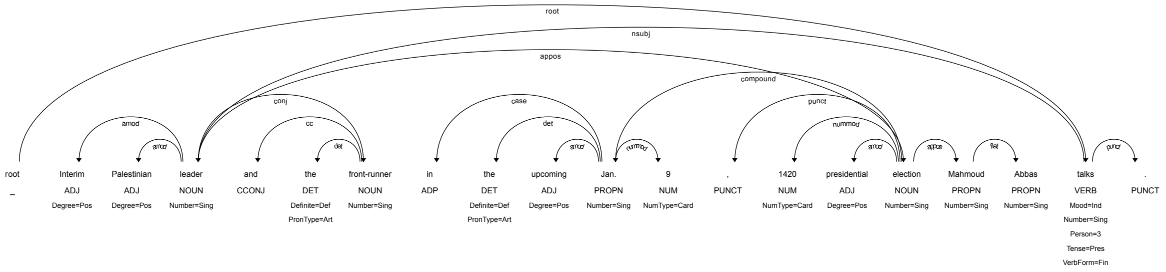

root
nsubj
appos
compound
conj
case
punct
amod
cc
det
nummod
amod
det
amod
nummod
amod
appos
flat
punct
root Interim Palestinian leader and the front-runner in the upcoming Jan. 9 , 1420 presidential election Mahmoud Abbas talks .
_ ADJ ADJ NOUN CCONJ DET NOUN ADP DET ADJ PROPN NUM PUNCT NUM ADJ NOUN PROPN PROPN VERB PUNCT
Degree=Pos Degree=Pos Number=Sing Definite=Def Number=Sing Definite=Def Degree=Pos Number=Sing NumType=Card NumType=Card Degree=Pos Number=Sing Number=Sing Number=Sing Mood=Ind
PronType=Art PronType=Art Number=Sing
Person=3
Tense=Pres
VerbForm=Fin

Figure 8: An example truncated dependency tree from cluster 6

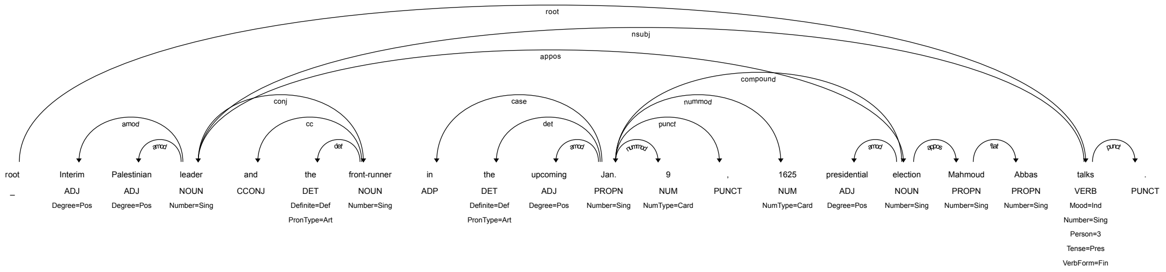

root
nsubj
appos
compound
conj
case
nummod
amod
cc
det
punct
amod
det
amod
nummod
amod
appos
flat
punct
root Interim Palestinian leader and the front-runner in the upcoming Jan. 9 , 1625 presidential election Mahmoud Abbas talks .
_ ADJ ADJ NOUN CCONJ DET NOUN ADP DET ADJ PROPN NUM PUNCT NUM ADJ NOUN PROPN PROPN VERB PUNCT
Degree=Pos Degree=Pos Number=Sing Definite=Def Number=Sing Definite=Def Degree=Pos Number=Sing NumType=Card NumType=Card Degree=Pos Number=Sing Number=Sing Number=Sing Mood=Ind
PronType=Art PronType=Art Number=Sing
Person=3
Tense=Pres
VerbForm=Fin

Figure 9: An example truncated dependency tree from cluster 7

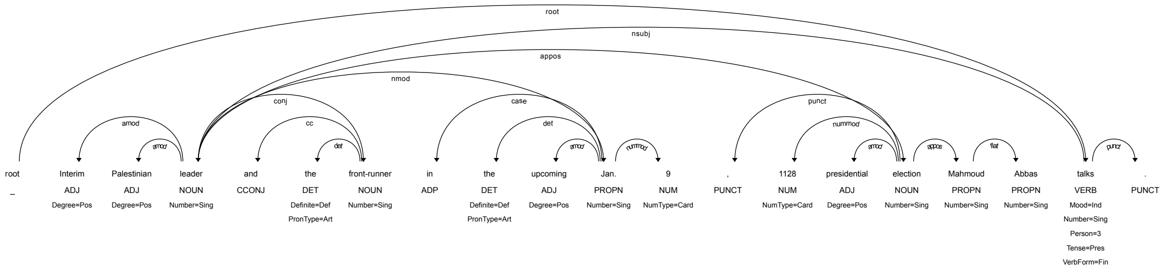

root
nsubj
appos
nmod
conj
case
punct
amod
cc
det
nummod
amod
det
amod
nummod
amod
appos
flat
punct
root Interim Palestinian leader and the front-runner in the upcoming Jan. 9 , 1128 presidential election Mahmoud Abbas talks .
_ ADJ ADJ NOUN CCONJ DET NOUN ADP DET ADJ PROPN NUM PUNCT NUM ADJ NOUN PROPN PROPN VERB PUNCT
Degree=Pos Degree=Pos Number=Sing Definite=Def Number=Sing Definite=Def Degree=Pos Number=Sing NumType=Card NumType=Card Degree=Pos Number=Sing Number=Sing Number=Sing Mood=Ind
PronType=Art PronType=Art Number=Sing
Person=3
Tense=Pres
VerbForm=Fin

Figure 10: An example truncated dependency tree from cluster 8

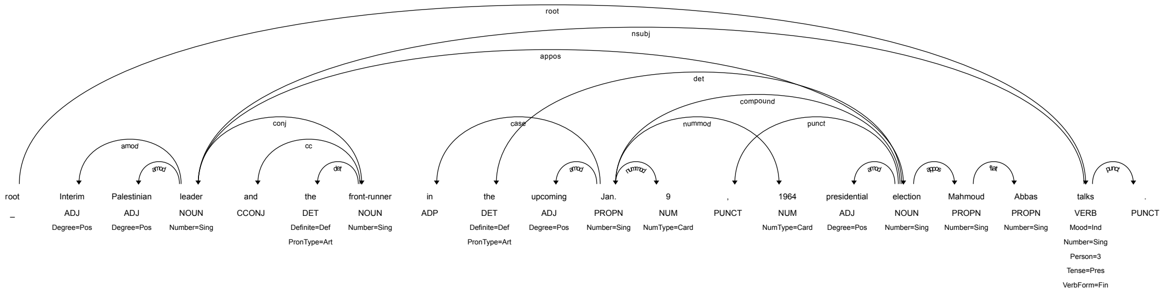


Figure 11: An example truncated dependency tree from cluster 9

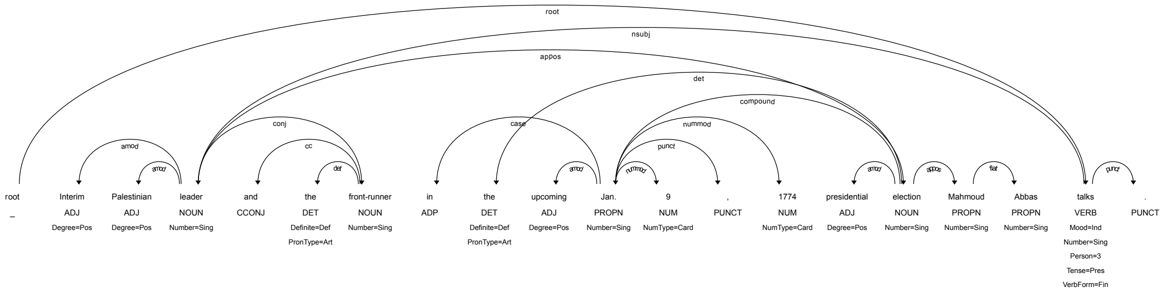


Figure 12: An example truncated dependency tree from cluster 10

Validation of Universal Dependencies by *regeneration*

Guy Lapalme
RALI-DIRO / Université de Montréal
C.P. 6128, Succ. Centre-Ville
Montréal, Québec, Canada, H3C 3J7
lapalme@iro.umontreal.ca

Abstract

We describe the design and use of a web-based system for helping the validation of English or French Universal Dependencies corpora by sentence regeneration. A symbolic approach is used to transform the dependency tree into a constituency tree which is then regenerated as a sentence in the original language. The comparison between regenerated sentences and the original ones from version 2.8 of Universal Dependencies revealed some annotation errors which are discussed and give rise to suggestions for improvement.

1 Introduction

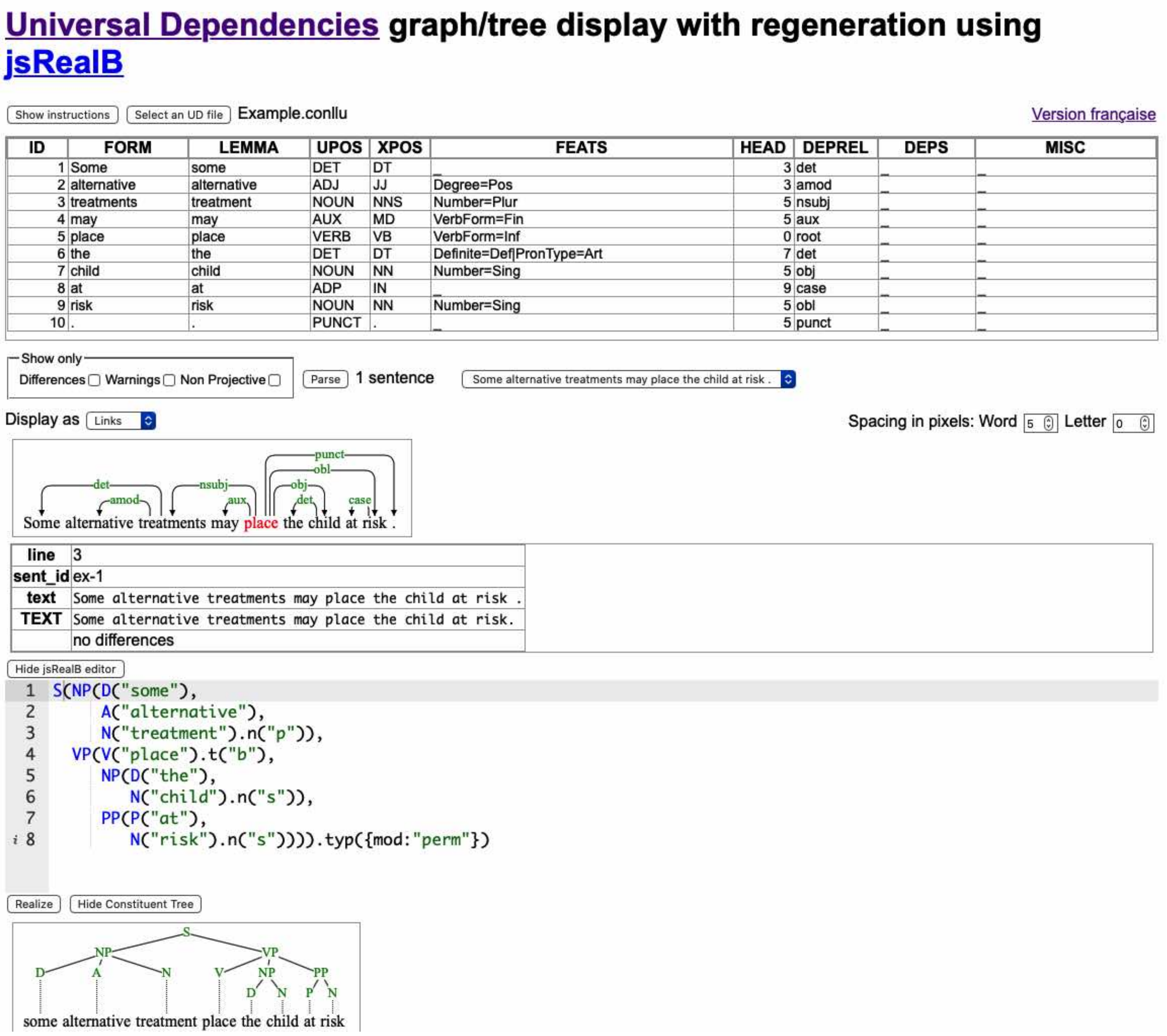
Universal Dependencies (de Marneffe et al., 2021) (UD) have been developed for comparative linguistics and are used in many NLP projects for developing parsers or machine learning systems for training and/or evaluation. The accuracy of these annotations is thus very important. Many of these dependency structure annotations are the result of manual revisions of automatic parses or mappings from other parsing formalisms (e.g. EWT was originally derived from the Penn Treebank annotations). They are often quite difficult to check manually as there are so many details to take into account. This paper describes an alternative way of looking at the UD data, using it to regenerate a sentence that can be compared with the original. As we will show in Section 3, regenerating from the source revealed small mistakes, most often omissions, in quite a few of these structures which are often considered as *gold standard*. It is indeed much easier to detect errors in a figure or in a generated sentence than in a list of tab separated lines.

This approach for helping detecting potential errors in annotations can be compared with the work of van Halteren (van Halteren, 2000) who compares the result of an automatic part-of-speech tagger with the ones in the corpus and highlights tokens in which disagreement occurs. Wisniewski (Wisniewski, 2018) detects all identical sequences of words that are annotated differently in a corpus. This is achieved by aligning, in Machine Translation parlance, the sentences with their annotation and then displaying both sequences highlighting their differences.

UDREGENERATOR (see Figure 1) is a web-based English and French realizer written in JavaScript, built using JSREALB[1]. Only the English realizer is shown here, but it is also possible to use it for checking the French corpora of UD. The table at the top shows the token fields of the selected UD in the menu with the corresponding dependency link structure in the middle.

A UD realizer might seem pointless, because most UD annotations are created from realized sentences either manually or automatically. As UD contains all the tokens in their original form (except for elision in some cases), the realization can be obtained trivially by listing the FORM in the second column of each line. Taking into account the tree structure, another baseline generator can be implemented by a recursive traversal of the UD tree by first outputting the forms of the left children, then the form of the head and finally the forms of the right children.

[1] http://rali.iro.umontreal.ca/rali/?q=en/jsrealb-bilingual-text-realiser



Universal Dependencies graph/tree display with regeneration using jsRealB

Example.conllu

ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	DEPS	MISC
1	Some	some	DET	DT	_	3	det	_	_
2	alternative	alternative	ADJ	JJ	Degree=Pos	3	amod	_	_
3	treatments	treatment	NOUN	NNS	Number=Plur	5	nsubj	_	_
4	may	may	AUX	MD	VerbForm=Fin	5	aux	_	_
5	place	place	VERB	VB	VerbForm=Inf	0	root	_	_
6	the	the	DET	DT	Definite=Def|PronType=Art	7	det	_	_
7	child	child	NOUN	NN	Number=Sing	5	obj	_	_
8	at	at	ADP	IN	_	9	case	_	_
9	risk	risk	NOUN	NN	Number=Sing	5	obl	_	_
10	.	.	PUNCT	.	_	5	punct	_	_

Show only
Differences ☐ Warnings ☐ Non Projective ☐

1 sentence

Some alternative treatments may place the child at risk .

Display as Links

Spacing in pixels: Word 5 Letter 0

line	3
sent_id	ex-1
text	Some alternative treatments may place the child at risk .
TEXT	Some alternative treatments may place the child at risk.
	no differences

```
S(NP(D("some"),
     A("alternative"),
     N("treatment").n("p")),
  VP(V("place").t("b"),
     NP(D("the"),
        N("child").n("s")),
     PP(P("at"),
        N("risk").n("s")))).typ({mod:"perm"})
```

Figure 1: Web page (http://rali.iro.umontreal.ca/JSrealB/current/demos/UDregenerator/UDregenerator-en.html) for exploring UD structures in a local file that is parsed to build a menu of their reference sentences in the middle of the page. Once a sentence is chosen, the fields of its tokens are displayed in the table at the top and the graph of its dependency links is displayed below the menu. A table below the graph shows information about this UD: the line number in the file, its sent_id and reference text (text), the regenerated text by JSREALB (TEXT). When there are differences between the expected text and realized text, they are highlighted (not shown here). The corresponding JSREALB expression is displayed in an editor that allows it to be changed and be re-realized. The constituency tree corresponding to the JSREALB expression is displayed at the bottom. Checkboxes are used to limit the sentences in the menu to those for which there are differences between the references text and the generated sentence, those for which JSREALB issued warnings or those with non-projective dependencies.

This method does not work for *non-projective* dependencies (Kahane et al., 1998) because then, words under a node are not necessarily contiguous. This property is used in our system to detect non-projective dependencies which account for about 4% of the dependencies in our corpora, which roughly corresponds to what was observed by Perrier (Perrier, 2021). But even for projective ones, different trees can be linearized in the same way. However we observed that, quite often, non-projective dependencies are a symptom of badly linked nodes that should be checked.

UDREGENERATOR realizes a sentence *from scratch* using only the lemmas and the morphological and syntactic information contained in the UD features and relations.

1.1 Constituency structure format

UDREGENERATOR transforms the UDs into a constituency structure format used as input for JSREALB (Molins and Lapalme, 2015), a surface realizer written in JavaScript similar in principle to SIMPLENLG (Gatt and Reiter, 2009) in which programming language instructions create data structures corresponding to the constituents of the sentence to be realized. Once the data structure (a tree) is built in memory, it is traversed to create the list of tokens of the sentence.

As is shown in Figure 1, the data structure is built by function calls whose names were chosen to be similar to the symbols typically used for constituent syntax trees[2]:

- **Terminal**: `N` (Noun), `V` (Verb), `A` (adjective), `D` (determiner) ...
- **Phrase**: `S` (Sentence), `NP` (Noun Phrase), `VP` (Verb Phrase) ...

Features, called *options*, that modify some properties are added to the structures using the dot notation. For terminals, they are used to specify the person, number, gender, etc. For phrases, the sentence may be negated or set to a passive mode; a noun phrase can be pronominalized. Punctuation signs and HTML tags can also be added.

For example, in the JSREALB structure of Figure 1, plural of `treatment` is indicated with the *option* `n("p")` where `n` indicates the number and `"p"` the plural. Agreements within the `NP` and between `NP` and `VP` are performed automatically, although this feature is not often used in this experiment because features on each token provide, *in principle*, the appropriate morphological information.[3]

The affirmative sentence is modified to use the *permission* modal using the property `{typ({"mod":"perm"})` to be realized by the verb `may`. The modification of a sentence structure is an interesting feature of JSREALB. Once the sentence structure has been built, many variations can be obtained by simply adding a set of options to the sentences, to get negative, progressive, passive, modality and some type of questions. For example, the interrogative form What may place the child at risk? could be generated by adding `"int":"was"` to the object given as parameter to `.typ()` at the end of the original JSREALB expression. This feature is not needed in this work, but it was used for creating questions from affirmative sentences to build a training corpus for a neural question-answering system or for creating negations for augmenting a corpus of negative sentences for training a neural semantic analyzer.

2 Building the Syntactic Representation

The first step, not described here, is straightforward: the CONLLU input format is parsed into a JavaScript data structure for the tokens. Each token keeps information from the UD fields such as `FORM`, `LEMMA`, `FEATS`, `DEPREL` and `MISC`. The `HEAD` field is used to build a list of pointers to the tokens on its left and an another list for the tokens to the right *children*. So starting from the root, it is possible to obtain the whole UD tree. Although the position of a node, given its `ID`, is not taken into account during realization, the positions of the children relative to their parent are kept intact. We do not take into account the *absolute* node positions, because our goal is to regenerate the sentence from the *relative* positions indicated by the UD relations.

[2] See the documentation `http://rali.iro.umontreal.ca/JSrealB/current/documentation/user.html?lang=en` for the complete list of functions and parameter types.

[3] Section 3 will show that, unfortunately, this is not always the case.

We now describe how a parsed UD is transformed into a Syntactic Representation (SR) which is used as input to jsRealB. The principle is to *reverse engineer* the UD annotation guidelines[4]. This is similar to the method described by Xia and Palmer (Xia and Palmer, 2001) to recover the syntactic categories that are *projected* from the dependents and to determine the extents of those projections and their interconnections.

Although this projection process is theoretically simple, there are peculiarities to take into account when it is applied between two predefined formalisms. In this case, the UD relations with features being associated with each token must be mapped into jsRealB constituents with options that are applied either to a terminal or a phrase. We now give more detail on the mapping process for generating words using the morphological information associated with tokens and for generating phrases from dependency relations.

2.1 Morphology

UD Terminals are represented in JavaScript as tokens with no children. They are mapped to *terminal* symbols in jsRealB. So we transform the JavaScript version of the UD notation to the SR one by mapping lemma and feature names. The following table gives a few examples:

JavaScript fields	SR

`"upos":"NOUN", "lemma":"treatment",`
`"feats":{"Number":"Plur"}`
	`N("treatment").n("p")`

`"upos":"VERB", "lemma":"lean",`
`"feats":{"Mood":"Ind","Tense":"Past"}`
	`V("lean").t("ps")`

`"upos":"PRON", "lemma":"its",`
`f̈eats":{"Gender":"Neut","Number":"Sing",`
`"Person":"3","Poss":"Yes","PronType":"Prs"}`
	`Pro("me").c("gen").pe("3")`
	`.g("n").n("s")`

As shown in the last example, we had to *normalize* pronouns to what jsRealB considers as its base form. In the morphology principles of UD[5], it is specified that *treebanks have considerable leeway in interpreting what canonical or base form means*. In some English UD corpora, the `lemma` of a pronoun is almost always the same as its `form`; it would have been better to use the tonic form. We decided to *lemmatize further* instead of merely copying the lemma as a string input to jsRealB so that verb agreement could eventually be performed. English UD do not seem to have a systematic encoding of possessive determiners such as `his` which, for jsRealB at least, should be POS-tagged as a possessive determiner. These are defined as pronouns in some sentences or determiners in others, we even found cases of both encodings occurring in the same sentence. As the documentation seems to favor pronouns,[6] we had to adapt our transformation process to deal with these *errors* as they occur quite often. This problem is less acute in the French UD corpora.

What should be a `lemma` is a hotly discussed subject on the UD GitHub[7], but there are still many *debatable* lemmas such as *an*, *n't*, plural nouns, etc. In one corpus, lowercasing has been applied to some proper nouns, but not all. We think it would be preferable to apply a more *aggressive* lemmatization to decrease the number of base forms to help further NLP processing that is often dependent on the number of different types. The lexica for jsRealB being sufficiently comprehensive for most current uses (34K lemmas for English and 53K lemmas for French), there are still unknown lemmas for specialized or informal contexts. Our experience shows that, most often, *unknown* lemmas are symptoms of errors in the lemma or the part of speech fields. Section 3.1 shows examples encountered in the corpora.

[4] `https://universaldependencies.org/guidelines.html`

[5] `https://universaldependencies.org/u/overview/morphology.html`

[6] `https://universaldependencies.org/en/feat/Poss.html` indicates that `his` can be marked as a possessive pronoun.

[7] `https://github.com/UniversalDependencies/docs/labels/lemmatization`

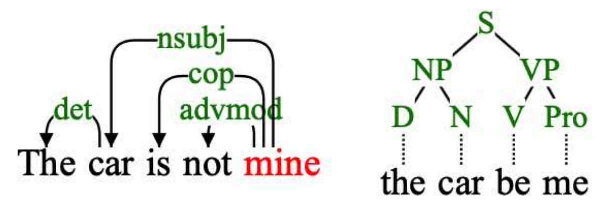
2.2 Translating the JavaScript notation of UD to Syntactic Representation

The goal is to map the JavaScript tree representation of the dependencies to a constituency tree that can be used by JSREALB to realize the sentence. According to the UD annotation guidelines, there are two main types of dependents: nominals and clauses, which themselves can be simple or complex.

The head of a Syntactic Representation is determined by the terminal at the head of the dependencies. The system scans dependencies to determine if the sentence is negative, passive, progressive or interrogative depending on whether combinations of `aux`, `aux:pass` with proper auxiliaries (possibly modals) or interrogative `advmod` are found. When such a combination is found, then these relations are removed before processing the rest. The appropriate JSREALB sentence `typ` will be added to the resulting Universal Dependencies. For example, in Figure 1, the auxiliary `may` is removed from the tree and the sentence is marked to be realized using the *permission* modal.

All dependencies are transformed recursively; as each child is mapped to a SR, children list are mapped to a list of SR. Before combining the list of Syntactic Representations into a JSREALB constituent, the following special cases are taken into account, for English sentences:

1. a UD with a copula is most often rooted at the attribute (e.g., `mine` in Figure 2), The constituency representation must be reorganized so that the auxiliary is used as the root of a verb phrase (`VP`): This reorganization could probably be simplified with the use of the Surface Syntactic Universal Dependencies formalism (SUD)[8] as input as it emphasizes syntax over semantics.



```
S(NP(D("the"),
     N("car").n("s")),
  VP(V("be").t("p").pe("3").n("s"),
     Pro("me").c("gen").pe("1")
              .g("n").n("s"))
 ).typ({neg:true})
```

Figure 2: On the left, the dependency tree corresponding to the sentence The car is not mine; the center shows the constituency tree after transformation with its JSREALB encoding on the right to realize the original sentence.

2. A verb at the infinitive tense is annotated in UD as the preposition `to` before the verb, so this preposition is removed before processing the rest of the tree, it is reinserted at the end;

3. An adverb (from `advmod` relation) is removed from processing the rest and added to the resulting `VP` at the end;

4. If the head is either a noun, an adjective, a proper noun, a pronoun or a number, it is processed as a nominal clause mapped to a `NP` enclosing all its children UD.

5. If the head is a verb: check if the auxiliary `will` is present, then a future tense option will be added to the verb; in the case of the `do` auxiliary, feature information (tense and person) is copied into the JSREALB options.

6. Otherwise, bundle Syntactic Representations into a sentence `S`, the subject being the first child and the `VP` being the second child.

7. Coordinate `VP`s and `NP`s must also be dealt in a special way because the way that JSREALB expects the arguments of a `CP` is different from the way coordinates are encoded in UD (see Figure 3) where the elements are joined by `conj` relations. in JSREALB, all these elements must be wrapped in a global `CP`, the conjunction being indicated once at the start.

[8] \sudurl{}

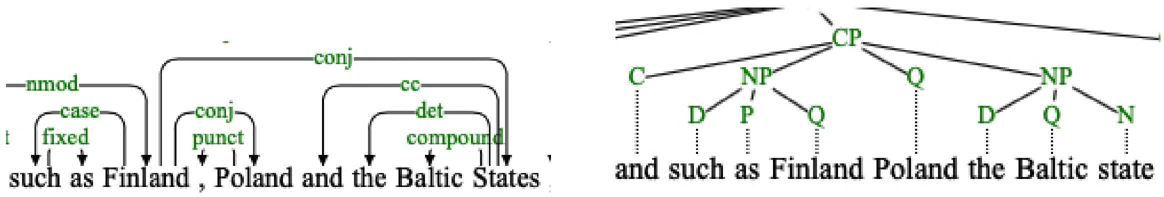

nmod
case
fixed
conj
conj
punct
cc
det
compound
such as Finland , Poland and the Baltic States
CP
C
NP
Q
NP
D
P
Q
D
Q
N
and such as Finland Poland the Baltic state

Figure 3: The graph at the left, a subgraph of the UD w02013093 in en_pud-ud-test.conllu, illustrates the UD encoding of coordinated nouns Finland, Poland and the Baltic States; the right part shows the constituency tree expected by JSREALB.

This exercise in transforming UD structures to JSREALB revealed an important difference in their level of representation. By design UD stays at the level of the form in the sentence, while JSREALB works at the constituent level. For example, in UD, negation is indicated by annotating `not` and the auxiliary elsewhere in the sentence, while in JSREALB the negation is given as an *option* for the whole sentence. So as shown above, the structure is checked for the occurrence of not and an auxiliary to generate the `.typ({neg:true})` option for JSREALB (see Figure 2); these dependents are then removed for the rest of the processing. Similar checks must also be performed for passive constructs, modal verbs, progressive, perfect and even future tense in order to *abstract* the UD annotations into the corresponding structure for JSREALB. It would be interesting to check if working with SUD (Gerdes et al., 2018) would simplify the transformation process into the dependency structure.

2.3 Working with French

As JSREALB can also be used for realizing sentences in French and that many UD are available in French, we adapted for French the methodology described in the previous section. For morphology, we changed the lemmas for pronouns and numerals. Fortunately, the *ambiguity* between pronouns and determiners seldom occurs in the French UD corpora, so this step was more straightforward. The transformation for clauses stays essentially the same as for English, except that there is no need to cater for the special cases for modals, future tense and infinitives.

3 Experiments

UDREGENERATOR can be used interactively,[9] but it can also be used as a NODE.JS module to process a corpus and print the differences between the original text and the regenerated one. We ran the NODE.JS version on the French and English corpora of UD (Version 2.8), the most recent at the time of writing.[10] UDREGENERATOR handled all sentences and is quite fast: about 1 millisecond per sentence on a commodity Mac laptop.

When all lemmas of UD structure appear in the JSREALB lexicon and are used with the appropriate features, UDREGENERATOR creates a tree and realizes the corresponding sentence. In other cases, JSREALB emits warnings so that the unknown words can either be corrected or added to the lexicon of JSREALB; in those cases, JSREALB inserts the *lemma* verbatim in the generated string which works out all right in English which is *not too morphologically rich*. But in many cases, these *erroneous lemmata* should be more closely checked. The tokens of the generated sentence are then compared with the tokens of the original text using the Levenshtein distance ignoring case and spacing. When there are differences, they are highlighted in the output or the display; the number of UD with differences are called *#diff* in the following tables. Differences can come from limitations of JSREALB (e.g., contractions, special word ordering that cannot be generated verbatim, non-projective dependencies) or from errors or underspecification of the part-of-speech, features or head field in the UD.

For this experiment, we consider generated sentences with non-projective dependencies as *errors*. In

[9] http://rali.iro.umontreal.ca/JSrealB/current/demos/UDregenerator/UDregenerator-en.html

[10] Using a previous version of this tool, we detected errors in Version 2.7 that were then corrected by the maintainers of the corpora once we raised the issues to them.

Corpus	type	#sent	#toks	#nPrj	#diff	#lerr	%regen	%terr	%nPrj
ewt	dev	2,001	25,149	44	987	219	51%	1%	2,2%
	test	2,077	25,097	41	970	174	53%	1%	2,0%
	train	12,543	204,584	462	6,780	1,698	46%	1%	3,7%
gum	dev	843	16,164	49	486	153	42%	1%	5,8%
	test	895	16,066	43	478	149	47%	1%	4,8%
	train	5,664	102,258	263	3,204	1,073	43%	1%	4,6%
lines	dev	1,032	19,170	102	664	228	36%	1%	9,9%
	test	1,035	17,765	67	645	257	38%	1%	6,5%
	train	3,176	57,372	254	2,040	702	36%	1%	8,0%
partut	dev	156	2,722	4	83	33	47%	1%	2,6%
	test	153	3,408	1	86	11	44%	0%	0,7%
	train	1,781	43,305	33	946	392	47%	1%	1,9%
pronouns	test	285	1,705	-	65	-	77%	0%	0,0%
pud	test	1,000	21,176	45	550	197	45%	1%	4,5%
Total		32,641	555,941	1,408	17,984	5,286	47%	1%	4,1%
sample		60	1,086	-	30		50%	0%	0,0%

Table 1: Statistics for the English UD corpora: for each corpus and type, it shows the numbers of sentences (#sent), tokens (#toks) and non-projective dependencies (#nPrj); the number of sentences that had at least one difference with the original (#diff); the number of tokens that had at least one lexical error (#lerr); the percentages of sentences regenerated exactly (%regen), of tokens in error (%terr) and of non-projective sentences (%nPrj). The next-to-last line displays the total of these values and the mean percentages over all sentences of the corpora. The last line shows the statistics for the sample that is studied more closely in Section 3.3.

the case of *legitimate* non-projectivity, the generated sentences are appropriate but with a different word order. But we also found quite a few non-projective dependencies caused by a single erroneous head link which can easily be fixed using the tree display showing crossing links. It might be interesting to explore generating the original word order by generating each token separately using only their lemma and features, but then we would lose the opportunity of checking links.

As we use a symbolic approach, we do not distinguish between the training, development and test splits of a corpus, we consider them as different corpora. This allows an overall judgment on what we feel to be the precision of the information in the UD. The last subsection provides a more detailed analysis of a representative sample of the corpora.

3.1 English corpora

Table 1 shows statistics on the 14 English corpora that comprise 32,641 sentences, of which 1,408 (4,1%) have non-projective dependencies and gave rise to 17,984 warnings. We did not use the three English ESL corpora because they do not provide any information about the lemma and the features of tokens, they only give the form and relation name.

Table 1 shows that on average about 47% of the sentences are regenerated exactly ignoring capitalization and spacing. Many of the differences are due to contractions (e.g. aint or he'll) for which JSREALB realizes the long form (is not or he will). There are two outliers: the `pronouns` corpus which uses a limited vocabulary and was manually designed to illustrate many variations of pronouns; in fact, we used it to design our pronoun transformations; the *lines* corpus has a high ratio of unknown lemmata, some of which are *dubious*: (collapsible|expandable), &amp;, course as an adverb, smile' and even wrote, which occurs 11 times or opened, 21 times.

Looking at the results, we found that one important source of differences was the fact that in many English corpora, person and number were not given as features of verbs except for the third person

United				
Corpus	**#occ**	**upos**	**lemma**	**feats**
EWT	93	`ADJ`	`United`	`Degree=Pos`
GUM	80	`VERB`	`Unite`	`Tense=Past,VerbForm=Part`
Lines	9	`PROPN`	`United`	`Number=Sing`
Partut	11	`PROPN`	`United`	
PUD	6	`PROPN`	`United`	`Number=Sing`

New				
Corpus	**#occ**	**upos**	**lemma**	**feats**
EWT	95	`ADJ`	`New`	`Degree=Pos`
GUM	80	`PROPN`	`New`	`Number=Sing`
Lines	21	`PROPN`	`New`	`Number=Sing`
Partut	3	`PROPN`	`New`	
PUD	7	`PROPN`	`United`	`Number=Sing`

Table 2: This table shows the different, and inconsistent across English corpora, part of speech, lemma and features associated with two common English words used in proper nouns. The second column gives the number of occurrences in each English corpus.

singular. There are about 11.5K instances of these in the EWT corpus[11], but none in the GUM corpus and about 9K in all other English corpora. As jsRealB uses the third person singular as default, the generated sentence comes out right most of the time, except when the subject is a pronoun at the first or second person or is plural.

We also discovered some inconsistencies between English corpora even for very common words. Table 2 shows occurrences of United used in United States, United Nations or United Kingdom and of New such as in New York, New England, New Delhi... In the previous version (2.7) of UD, all United had been tagged as `PROPN`.

Given the fact that the jsRealB lexicon does not include the adjective United (with a capital U) or the verb Unite also with a capital, this raised warnings. A similar problem occurred for the adjective New used in New Year, New Left for which some occurrences are adjectives and others are part of a proper noun. jsRealB lexicon does not contain these lemmata with a capital. This may seem anecdotical, but it occurs quite frequently and is typical of inconsistency problems.

Another source of warnings is the fact that some words are tagged dubiously: there are strange conjunctions such as: of (264 occurrences), in (181), by (162), with (142), on (99) ...

Although POS tags are consistent most of the time within a corpus, this is not the case between corpora of the same language, especially for a *non-low resource* language such as English, so some care should be used when combining these corpora in a learning scheme for a given language, unless the learning scheme does not care about number, person and POS tags.

In order to limit the number of warnings, we decided to add a few *dubious* lemmas[12]:

- best and better were added as lemmas, although we think that the appropriate lemma should be good or well specifying the `Degree` feature: superlative (`Sup`) or comparative (`Cmp`).

- & was added as a lemma for a conjunction, but it should be and.

- in formal English, adjective and nouns corresponding to nationalities start with a capital letter (e.g. American or European), but we also had to accept the lowercase form as lemma for these.

[11] Following our suggestion, most of these features have been added to EWT in version 2.9

[12] Many of these cases, have been corrected in version 2.8 of some corpora, namely EWT, following our remarks about this problem on version 2.7

Corpus	type	#sent	#toks	#nPrj	#diff	#lerr	%regen	%terr	%nPrj
fqb	test	2,289	23,901	75	1,321	682	42%	3%	3,3%
gsd	dev	1,476	35,707	60	702	522	52%	1%	4,1%
	test	416	10,013	17	233	147	44%	1%	4,1%
	train	14,449	354,529	587	6,670	4,971	54%	1%	4,1%
partut	dev	107	1,870	2	64	64	40%	3%	1,9%
	test	110	2,603	1	62	68	44%	3%	0,9%
	train	803	24,122	49	506	463	37%	2%	6,1%
pud	test	1,000	24,726	17	445	460	56%	2%	1,7%
sequoia	dev	412	9,999	10	175	190	58%	2%	2,4%
	test	456	10,044	9	204	182	55%	2%	2,0%
	train	2,231	50,505	47	992	915	56%	2%	2,1%
spoken	dev	919	9,973	73	400	252	56%	3%	7,9%
	test	743	9,968	80	220	300	70%	3%	10,8%
	train	1,175	15,031	103	509	347	57%	2%	8,8%
Total		26,586	582,991	1,130	12,503	9,563	53%	2%	4,3%
Sample		60	1,233	3	37	24	38%	2%	5,0%

Table 3: Statistics for the French UD corpora: for each corpus and type, it shows the numbers of sentences (#sent), tokens (#toks) and non-projective dependencies (#nPrj); the number of sentences that had at least one difference with the original (#diff); the number of tokens that had at least one lexical error (#lerr); the percentages of sentences regenerated exactly (%regen), of tokens in error (%terr) and of non-projective sentences (%nPrj). The next-to-last line displays the total of these values and the mean percentages over all sentences of the corpora. The last line shows the statistics for the sample that is studied more closely in Section 3.3.

3.2 French corpora

The 14 French UD corpora (see Table 3) provide 26,586 sentences of which 1,130 (4,3 %) have non-projective dependencies. UDREGENERATOR regenerates about 53% of the sentences, which is slightly more than for the English corpora, but the overall statistics are similar between French and English.

As for English, many of the warnings were generated by *strange* part of speech tags: comme tagged as a preposition instead of adverb or conjunction (796 occurrences), puis as conjunction instead of adverb (225 occurrences). There were a number of incomplete or erroneous lemmata. Here are a few examples across all French corpora:

bad part of speech : certain (343 times) is a determiner instead of an adjective; comme (796 times) is a preposition instead of a conjunction;

orthographic error in lemma : region (14 times) instead of région, inégalite (4 times) instead of inégalité, pubblicitaire (5 times) instead of publicitaire;

bad lemma : humains (6 times), performances (5 times), normes (4 times), financements, intactes or ressources whose lemma should be singular.

This is a good illustration of how UDREGENERATOR can help `improve` UD information.

In both French and English corpora, we found a few instances of bad `head` links for which regeneration produces words in the wrong order. We noticed that most often this occurs in non-projective dependencies, the tree representation is particularly useful for checking these as there are crossing arcs. This is why the system flags these so that they can be identified more easily and checked.

3.3 Sample corpora

In order to get a more precise appraisal of the quality of the UD information, we studied in detail a sample of 10 sentences from the 6 English and French test corpora for which we used UDREGENERATOR to recreate the original sentences.[13] The percentages on the last line of Tables 1 and 3 show that these samples have roughly the same characteristics as the whole corpus from which they were taken, except for the fact that there are no non-projective dependencies in the English sample.

This experiment shows that JSREALB has an almost complete coverage of English and French grammatical constructs found in the corpora, except for some specialized terminology which can be easily added to the lexicon or given as quoted words that will appear verbatim in the output. We encountered only 12 unknown tokens over 1,086 in English (e.g. shippeddate, luncheonnette as noun, related or numismatic as adjective) and 4 unknown tokens over 1233 in French (e.g. boxeuse as noun or déclassifier as verb). In some cases (7 for English and 5 in French) JSREALB could not reproduce the exact order of some of the words in a sentence: e.g., when an adverb is inserted within a conjugated modal (e.g. would never use) or because of non-projective dependencies. In all cases, the sentences kept their original meaning.

In English, 24 sentences were reproduced verbatim, without any modification either to the UD coding or the generated JSREALB expression. There were 5 cases of contractions (e.g., doesn't instead of does not, I've instead of I have) that JSREALB does not generate. Three cases of limitations of JSREALB because of modals being applied to noun phrases because of the transformation process limits. But we found 23 tokens (over 1,086) for which there were errors or omissions in either the part of speech tags (UPOS), features or lemma (e.g. follow as adjective, Sir or Council as noun, with or of as conjunction). Those are very small numbers computed over only 60 sentences, the whole corpus being 490 times greater.

We also experimented with 60 sentences sampled from French corpora with the following results: 26 were regenerated verbatim without any intervention. 48 tokens (over 1233) had errors or omissions in either the part of speech tags (UPOS), features or lemma (e.g. mot as adjective, octobre and expire as a feminine noun, but voile (in the sense of *sail*) as a masculine noun even though the attribute is feminine). There were 5 cases of word ordering in some part of a sentence, most because of non-projective dependencies, a case of an incomplete sentence and an unusual encoding of coordination. The encoding of pronouns is especially delicate because different corpora do not use the same conventions. A case of JSREALB limitations was encountered for the verb *pouvoir*: je peux at interrogative form should be realized as puis-je and not peux-je.

This is an experiment over a very small sample (0,23%) of sentences from the French and English corpora, but it shows the need to recheck the information in UD as it is often used as a gold standard and sometimes even used as a *mapping* source for other lower-resourced languages. We also showed that UDREGENERATOR it can be a useful tool for pointing out some eventual problems in the annotation of tokens and relations.

4 Conclusion

This work made us realize that UD corpora, while being a source of useful linguistic information, would benefit from a check by trying to regenerate the sentences from the provided annotation. We are not aware of any previous attempt to perform such an experiment. Sentence regeneration is not foolproof because different feature combinations can produce the same sentences, but we showed that in some cases it helps to pinpoint discrepancies between what is specified and the expected outcome. UDREGENERATOR is far from perfect, but it proved to be a convenient tool for doing some sanity checking on the lemma, part of speech, features and head fields. We hope that this work will help improve the precision of the wealth of useful information contained in UD corpora.

We only experimented with the French and English UD corpora because the text realizer we used only deals with these languages, but it showed the potential of detecting errors even in so-called *high resource* languages whose annotation is often considered as *golden*. It would be interesting to apply the same technique using a text realizer in other languages. Should a full-text realizer not be available for this target language, we conjecture that already checking the tokens with a conjugation or declension

[13] These sample corpora including the equivalent JSREALB expressions are available at http://rali.iro.umontreal.ca/JSrealB/current/demos/UDregenerator/UD-2.8/sample/

tool might already be useful to detect some *interesting* or dubious cases. The appendix describes briefly a language independent UD exploration tool that we used to pinpoint recurrent patterns that might be symptoms of errors.

Acknowledgements

We thank François Lareau from OLST in the Linguistics Department of Université de Montréal for many fruitful discussions and suggestions about this tool and for his long-standing and continued interest in JSREALB.

References

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

Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 90–93, Athens, Greece, March. Association for Computational Linguistics.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2018. SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In *Universal Dependencies Workshop 2018*, Brussels, Belgium, November.

Sylvain Kahane, Alexis Nasr, and Owen Rambow. 1998. Pseudo-projectivity, a polynomially parsable non-projective dependency grammar. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 646–652, Montreal, Quebec, Canada, August. Association for Computational Linguistics.

Paul Molins and Guy Lapalme. 2015. JSrealB: A bilingual text realizer for web programming. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 109–111, Brighton, UK, September. Association for Computational Linguistics.

Guy Perrier. 2021. Étude des dépendances syntaxiques non projectives en français. *Revue TAL*, 62(1).

Hans van Halteren. 2000. The detection of inconsistency in manually tagged text. In *Proceedings of the COLING-2000 Workshop on Linguistically Interpreted Corpora*, pages 48–55, Centre Universitaire, Luxembourg, August. International Committee on Computational Linguistics.

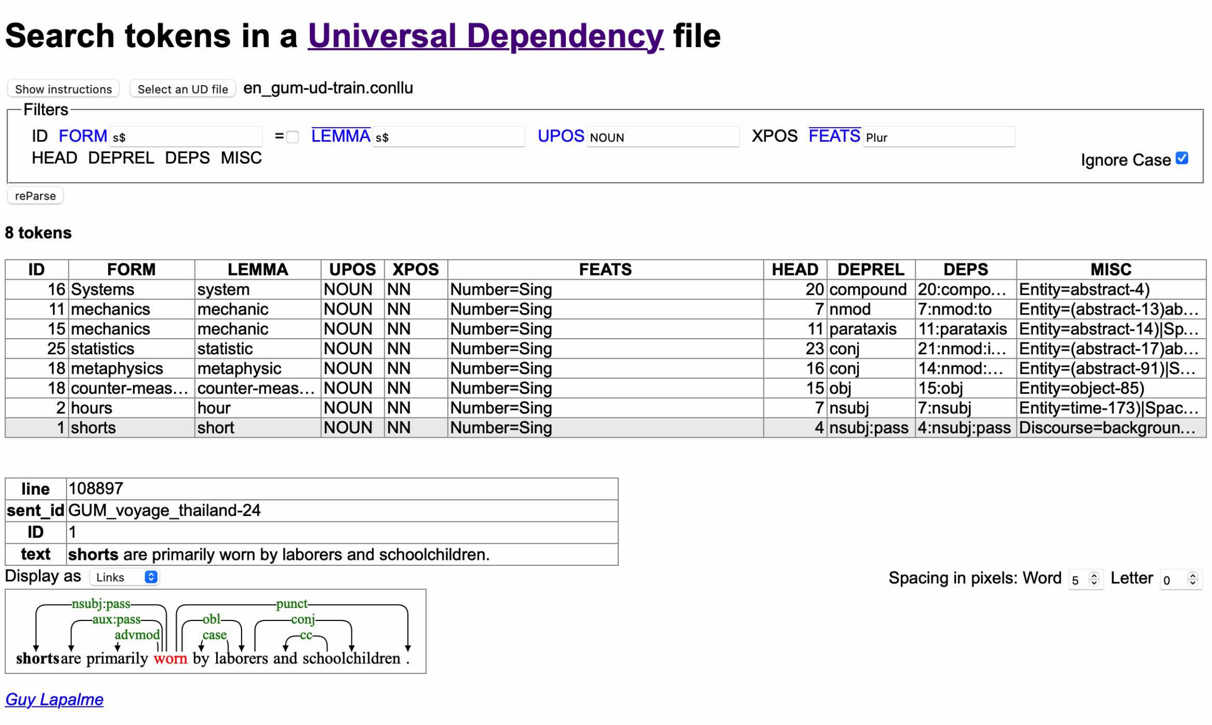
Guillaume Wisniewski. 2018. Errator: a tool to help detect annotation errors in the Universal Dependencies project. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Fei Xia and Martha Palmer. 2001. Converting dependency structures to phrase structures. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–5. Association for Computational Linguistics.

Appendix: Searching for combinations of tokens

UDREGENERATOR can identify some errors or missing features in a given UD, but we found interesting to search in a file of UDs if this combination exists in others UD. Many researchers use `grep` or string searching of a text editor or even special purpose scripts to check for specific combinations of features for a given token. None of these combinations are foolproof, but they can easily be checked once they are identified and they can then be modified in the original file if needed. To help identify these types of feature combinations, we have set up a web page[14] to search in local UD files. Each UD field can be matched for a regular expression or its negation. It is also possible to check if a `FORM` is the same or different from the `LEMMA`. All tokens in the file that match these conditions are displayed in a table, in which it is possible to select one to get the sentence in which it occurs, the identification of the sentence (`sent_id`) and the line number in the file. This tool is not as sophisticated as `Grew-match`[15] which defines a pattern language to allow also searching for combinations of links.

As UDGREP does not use any language specifics, it can be used to find patterns on UDs in any language. Patterns entered by the user are, of course, language specific.



Search tokens in a Universal Dependency file

en_gum-ud-train.conllu

Filters

ID FORM s$ =□ $\overline{\text{LEMMA}}$ s$ UPOS NOUN XPOS $\overline{\text{FEATS}}$ Plur
HEAD DEPREL DEPS MISC

Ignore Case ☑

8 tokens

ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	DEPS	MISC
16	Systems	system	NOUN	NN	Number=Sing	20	compound	20:compo…	Entity=abstract-4)
11	mechanics	mechanic	NOUN	NN	Number=Sing	7	nmod	7:nmod:to	Entity=(abstract-13)ab…
15	mechanics	mechanic	NOUN	NN	Number=Sing	11	parataxis	11:parataxis	Entity=abstract-14)|Sp…
25	statistics	statistic	NOUN	NN	Number=Sing	23	conj	21:nmod:i…	Entity=(abstract-17)ab…
18	metaphysics	metaphysic	NOUN	NN	Number=Sing	16	conj	14:nmod:…	Entity=(abstract-91)|S…
18	counter-meas…	counter-meas…	NOUN	NN	Number=Sing	15	obj	15:obj	Entity=object-85)
2	hours	hour	NOUN	NN	Number=Sing	7	nsubj	7:nsubj	Entity=time-173)|Spac…
1	shorts	short	NOUN	NN	Number=Sing	4	nsubj:pass	4:nsubj:pass	Discourse=backgroun…

line	108897
sent_id	GUM_voyage_thailand-24
ID	1
text	**shorts** are primarily worn by laborers and schoolchildren.

Display as Links

Spacing in pixels: Word 5 Letter 0

Guy Lapalme

Figure 4: Identification of *curious* English nouns that end with `s`, but not their lemma and that do not contain *Plur* in their features. A colored field name shows a regular expression that should match the field, a complemented name (with an overbar) shows a regular expression that should not match. The identified tokens are shown in a table in which it is possible to select a cell to show the context of this token: sentence with the token highlighted, the id of the sentence and its line number in the file. The dependency graph of the sentence is also shown.

[14] Available at `http://rali.iro.umontreal.ca/JSrealB/current/demos/UDregenerator/UDgrep.html`

[15] `http://match.grew.fr`

A Universal Dependencies corpus for Ligurian

Stefano Lusito
Universität Innsbruck
Institut für Romanistik
Innsbruck, Austria
stefano.lusito@uibk.ac.at

Jean Maillard
University of Cambridge
Dept. of Computer Science and Technology
Cambridge, United Kingdom
jean@maillard.it

Abstract

Ligurian is a minority Romance language spoken in the homonymous region of Northern Italy and the Principality of Monaco, amongst others. In this paper we present the first Universal Dependencies treebank for Ligurian, consisting of 316 sentences and 6 928 tokens, extracted from a wide variety of sources to reflect variation in syntax and register.

Along with the corpus, we contribute a short analysis of the varieties and spelling systems of Ligurian, as well as a set of recommendations and annotation guidelines for certain constructions with non-trivial analyses. We hope that these will serve as a foundation for further research, to encourage the development of NLP technologies for a language that has so far been under-served.

1 Introduction

Ligurian is a minority Romance language originating from the Northern part of the Italian peninsula, considered to be "definitely endangered" by UNESCO.[1] In spite of its relatively extensive usage throughout the centuries, no methodical corpus whatsoever exists for Ligurian, and no advanced NLP technologies have been developed for it.

Universal Dependencies (UD) (Nivre et al., 2016) is a cross-lingual framework for consistent annotations of parts-of-speech, morphological features, and syntactic dependencies. The project aims to facilitate the development of parsing technologies, enabling the use of techniques such as cross-lingual transfer.

In this paper we present the first ever digital corpus of Ligurian,[2] consisting of 316 sentences annotated according to the UD framework. We also contribute an analysis of the current state of the language, including its varieties and spelling system, and provide recommendations to serve as a foundation for future research.

The creation of a UD treebank for Ligurian enables the development of parsers and taggers for it, unlocking NLP technologies as well as software which is fundamental for linguistic research, such as advanced search tools for corpus linguistics (Guillaume, 2019).

The complete lack of technological support for Ligurian is – we believe – partly to blame for its endangered status. With this project we hope to encourage further research in the language and the development of NLP tools for it, in the hope of playing a small role in helping reverse a course which could otherwise lead to its complete disappearance.

2 The Ligurian language

2.1 Definition of *Ligurian*

Ligurian denotes the ensemble of Romance varieties traditionally spoken within the homonymous region of Liguria in Northern Italy. In its local forms, it is also the historical language of the Principality of Monaco as well as of the Tabarkin communities of Southern Sardinia, amongst others (Toso, 2003a; Toso, 2001; Toso, 2003b).

[1] http://www.unesco.org/languages-atlas/

[2] Available at https://github.com/UniversalDependencies/UD_Ligurian-GLT/.

Despite being traditionally associated to the so-called Gallo-Italic Romance dialects (Ascoli, 1876), the Ligurian varieties distinguish themselves from the other members of that group (Piedmontese, Lombard, Emilian and Romagnol) for their characters of conservatism and, at the same time, innovation in their evolution from vulgar Latin (Toso, 1995, p. 30). In this respect, it was the pioneering work of Diez (1836, p. 86) to first identify Ligurian as the transition area between North and Central Italian dialects.

The more recent division of Ligurian Romance varieties (Toso, 2002) distinguishes between a central, linguistically dynamic area (known as "Genoese" in the literature – *zeneise* in Ligurian) and marginal zones which have not received many of the innovative traits spread out from the central zone to a considerable portion of the region.[3] The Genoese dialects – whose extension includes the whole coastline from Noli to Framura and a sizeable portion of the corresponding inland region – cover more than a third of the administrative region's surface (encompassing many of the main urban areas) and represent by far the most widespread Ligurian variety as for number of speakers.

2.2 Role of Genoese and its literary production

In fact, Genoese is nowadays the only Ligurian dialect with a written corpus – mainly literary – which continuously stretches from the 13th century to the present day (Toso, 2009). Being traditionally considered the most prestigious Ligurian variety, it has also historically served as the *koinè* language for speakers of other Ligurian dialects – the usage of Italian having reached general oral diffusion only in the second half of the last century[4] – and functions still today as the Ligurian reference dialect when no particular diatopic information is required or specified (Toso, 1997).

2.3 Genoese spelling system

The long history of Genoese as a written tongue has led to the development of a spelling system which has evolved over the years together with the language itself (Toso, 2009, p. 27-32).

The influence of a relatively modest, but high quality literature among those who usually write in Genoese for public purposes is still such that all the main features of its traditional spelling system are generally accepted (e.g. ⟨o⟩ for [u], ⟨u⟩ for [y], ⟨æ⟩ for [ɛ(ː)] or ⟨x⟩ for [ʒ]). Nevertheless, the freedom allowed by the lack of both state recognition and prescriptive institutions still results in several disputed aspects, such as the writing of pre-tonic double consonants, always pronounced as singleton (e.g. *accattâ* vs. *acatâ* [akaˈtaː] 'to buy') and vowel-length markers, especially when a long vowel comes before the main stress of a word (e.g. *mäveggia* vs. *mâveggia* [maːˈved͡ʒˑa] 'wonder'). While, on the one hand, this situation leaves the field open to stimulating debates among the speakers, on the other it can sometimes generate confusion for the general public and even jeopardise meritorious projects. An illustrative case is the Ligurian edition of Wikipedia,[5] where the lack of uniform spelling guidelines (along with the use of a multitude of different local dialects) leads to a disorganised appearance (Lusito, 2021).

Driven by the aim to find a possible solution to these issues, a slight reform of Genoese spelling was recently proposed by a diverse group of journalists, writers, and academics (Acquarone, 2015). It has been adopted by the main Ligurian newspaper for its Ligurian-language columns (*Il Secolo XIX*), the book series *E restan forme* (poetry) and *Biblioteca zeneise* (prose)[6], the magazine *O Stafî* as well as the research project *GEPHRAS* currently running at the University of Innsbruck.[7]

Since the texts collected in this corpus come in large part from some of the aforementioned sources, this is also the spelling system adopted in this work.

[3] An in-depth outline of the evolutionary differences and features of the Ligurian dialects is to be found in Toso (1995, p. 30-42).

[4] Estimates for the percentage of people with an adequate knowledge of Italian at the time of the political unification of the country (1861-1870) range between 2.5% (De Mauro, 1991) and 9.5% (Castellani, 1982).

[5] https://lij.wikipedia.org/

[6] Respective pubishers Zona (http://www.editricezona.it/) and De Ferrari (https://www.deferrarieditore.it/).

[7] https://romanistik-gephras.uibk.ac.at/

Genre	**Documents**	**Paragraphs**	**Sentences**	**Tokens**
Fiction	4	19	76	2 216
News	2	7	59	1 472
Bible	1	13	46	1 241
Grammar examples	2	—	77	851
Wikipedia	2	8	18	754
Spoken	1	20	40	394
Total	12	67	316	6 928

Table 1: Composition of the Universal Dependencies corpus for Ligurian. *Documents* refer to chapters for the Fiction and Bible genres, and articles for the News and Wikipedia genres.

2.4 General Ligurian syntactic features

As already mentioned, the phonology, morphology, and syntax of Ligurian show features in the middle between those of North Italian dialects, on the one hand, and Tuscan and the South Italian ones, on the other.

Following Forner (1997, p.250-252), among some of the main features in contrast with standard Italian we find:

1. the presence of subject clitics,
2. compound demonstrative pronouns (although one-word pronouns also exist: *veuggio sto chì* besides the less frequent *veuggio questo* ‘I want this one’)[8],
3. bicomposed verbs, especially to express direction (*dâ quarcösa inderê* ‘to give something back’, Italian ‘restituire qualcosa’, *piccâ drento à quarcösa* / *quarchedun* ‘to crash against something / somebody’, Italian *scontrare qualcosa* / *qualcuno*), and
4. periphrastic structures to create progressive forms, with different possibile solutions (for ‘I am working’ one could use a construction with verb, adverb and infinitive, like *son derê à travaggiâ* or *son apreuvo à travaggiâ*, or a cleft sentence like *son chì che travaggio*; Italian has *stare* followed by a verb in gerund form instead: *sto lavorando*).[9]

3 Corpus development

3.1 Collection

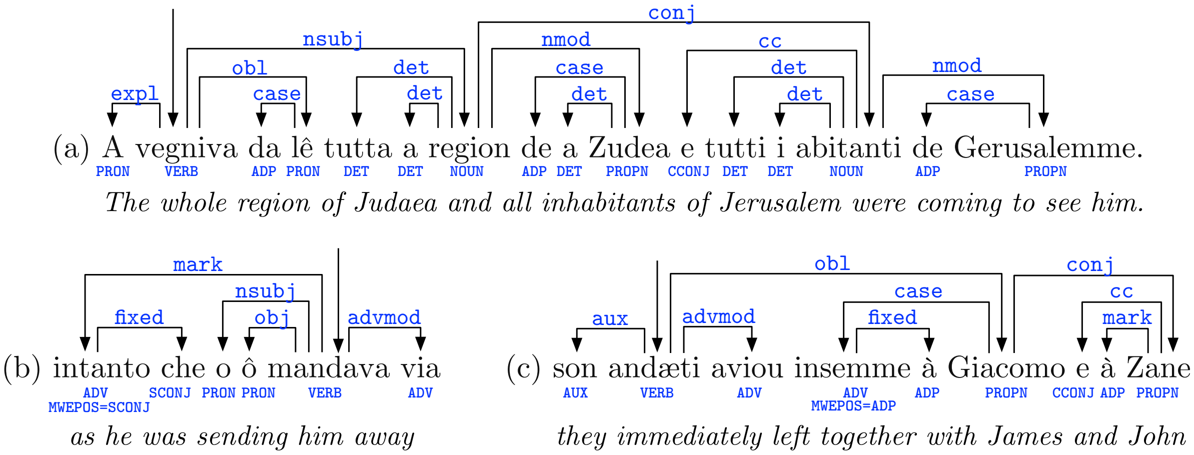
The texts included within the corpus (see Table 1) cover several genres and have been extracted from the most varied sources, in order to reflect variation in syntax and register. All texts were already written according to the aforementioned spelling system, which was maintained with minimal interventions to increase uniformity among them. The largest category, fiction, consists of four excerpts from three texts by contemporary authors or translators (Lusito, 2020; Toso, 2018; Iacopone, 2017). We also include one news and one magazine article (Canessa, 2016; Toso, 2020); an excerpt from a translation of the Gospel of Mark (Toso, 2019); two articles from the Ligurian edition of Wikipedia; a number of example sentences from a Genoese grammar book (Toso, 1997) selected to demonstrate a variety of characteristic syntactic constructions; and the transcript of a short comedy sketch, expressly conceived to be broadcast on the radio, but which accurately reflects oral usage. Finally, we include translations of the 20 example sentences making up the Cairo CICLing Corpus,[10] a multilingual parallel treebank of short sentences.

The relatively low fraction of texts coming from Wikipedia – a common source of content for textual corpora – is due to its inconsistent use of orthography and dialects, as mentioned in Section 2.3, as well as to quality issues with some of its contents, which appear to be written by novice language learners (Lusito,

[8] The non-marked Italian respective form is *voglio questo*. The construction comprising the adverb – *voglio questo qui* – is possible; in that case, if necessary, Ligurian would use a cleft sentence to mark the focus: *l’é sto chì che veuggio*.

[9] All the Ligurian examples are in Genoese.

[10] `https://github.com/UniversalDependencies/cairo`



conj
nsubj
nmod
obl
det
cc
det
nmod
expl
case
det
case
det
det
case
det
(a) A vegniva da lê tutta a region de a Zudea e tutti i abitanti de Gerusalemme.
PRON VERB ADP PRON DET DET NOUN ADP DET PROPN CCONJ DET DET NOUN ADP PROPN
The whole region of Judaea and all inhabitants of Jerusalem were coming to see him.
mark
nsubj
fixed
obj
advmod
(b) intanto che o ô mandava via
ADV SCONJ PRON PRON VERB ADV
MWEPOS=SCONJ
as he was sending him away
obl
conj
case
cc
aux
advmod
fixed
mark
(c) son andæti aviou insemme à Giacomo e à Zane
AUX VERB ADV ADV ADP PROPN CCONJ ADP PROPN
MWEPOS=ADP
they immediately left together with James and John

Figure 1: Some examples of annotated Ligurian sentences drawn from the corpus.

2021).

3.2 Annotation

The annotation was performed entirely manually by two trained linguists – both native Ligurians and intimately familiar with the language – using the CoNLL-U Editor tool (Heinecke, 2019). A two-step process was used: a first pass of annotations was followed by discussions, after which both annotators went back separately over their initial annotations. Inter-annotator agreement was measured on a sample of 60 sentences from the fiction domain, which was found by the annotators to be by far the most complex and difficult part of the corpus to label. Agreement, calculated with Cohen's kappa, was 0.97 for POS tags and 0.84 for labelled dependencies in the first round. After the second round of annotations, agreement increased to 0.99 for POS tags and 0.97 for dependencies. One of the most frequent sources of disagreement involved the clitics *ghe* and *ne*, which are traditionally treated as adverbs but can sometimes be seen as demonstrative pronouns (Toso, 1997).

We discuss here the key aspects of the guidelines developed for the UD annotation of Ligurian, focussing on the analyses which might not be immediately self-evident. Some of these are exemplified in Fig. 1.

Tokenisation Tokenisation is performed by whitespace and punctuation, analogously to other Romance languages. Multi-word tokens are used for clitics (*andemmosene* → *andemmo se ne*, 'let us go away from here'; *pensâghe* → *pensâ ghe* 'to think about it';) as well as for adpositions fused with articles (*in sciô* → *in sce o* 'on the'; *do* → *de o* 'of the'; *a-a* → *à a* 'at the').

Articles They are marked by `PronType=Art`, and can be definite (`Definite=Def`, *o, a, l', e, i*) or indefinite (`Definite=Ind`, *un, unna, do, da, di, de*). They have grammatical gender and number.

Adjectives These can have grammatical gender, number, and degree. Comparative and superlatives which differ from their positive form (*megio, pezo*) are marked `Degree=Cmp`. Absolute superlatives (*braviscima* 'very good') are marked `Degree=Abs`. All other cases are denoted by the absence of the `Degree` feature.

Numerals Ordinal numerals are tagged `ADJ` with `NumType=Ord`, and have gender and number. Cardinals are tagged `NUM` with `NumType=Card`. Some cardinals – *un/uña* 'one', *doî/doe* 'two', *trei/træ* 'three' and their composites (e.g. *vintidoî* 'twenty-two' masc., *vintidoe* 'twenty-two' fem.) – also have grammatical gender.

Auxiliaries We mark as `AUX` the copular verbs *ëse* ('to be') and *stâ* (functionally equivalent to the Spanish 'estar') when functioning as copula; the passive auxiliaries *ëse* and *an(d)â* 'to go'; the tense auxiliaries *ëse* and *avei* 'to have'; and the passive auxiliary *vegnî* and *an(d)â*. We also mark as `AUX` the modals *poei* ‘to be able to’, *dovei* ‘to have to’, *voei* ‘to want’, and *savei* ‘to know’, following the treatment of analogous verbs in Universal Dependencies treebanks of other Romance languages.

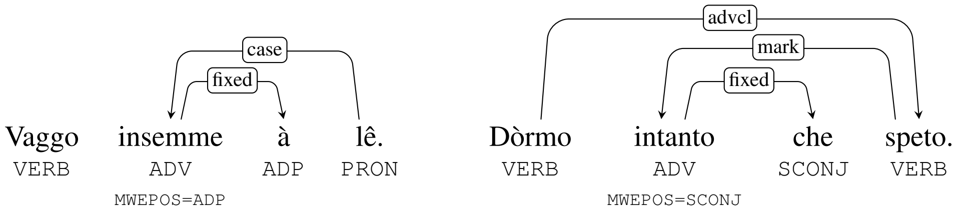


Figure 2: Grammaticalised multi-word expressions. *I go along with her* and *I sleep while I wait.*

Multi-word expressions Expressions which have undergone grammaticalisation are joined with the `fixed` relation. When the part-of-speech annotation of the head token does not match that of the expression as a whole, we use the additional annotation `MWEPOS` (in the `MISC` column of the CoNLL-U format) to indicate the part-of-speech of the expression as a whole. Examples of multi-word expressions include conjunctions (*intanto che* ‘while’, *de za che* ‘since’), adverbs (*de longo* ‘always’, *in derê* ‘behind’), and prepositions (*in sce* ‘on’, *in cangio de* ‘instead of’).

Clitic doubling In Ligurian, subject pronominal doubling is normally mandatory for the third person singular (*o Gioan o mangia* ‘Gioan eats’) and in some dialects for the third person plural (*i mæ amixi i mangian* ‘my friends eat’). In sentences where both the clitic and the lexical subject appear, the former is marked `expl`.

Ghe* and *ne The clitic *ghe*, when not acting as personal pronoun, is traditionally seen as an adverb, but can in many cases be interpreted as a demonstrative pronoun: *cöse ghe pòsso fâ?* (‘what can I do about that?’). The particle *ne* represents an analogous case. Due to the subtlety of these distinctions, it was decided that these clitics, when not acting as personal pronouns, would be tagged `ADV`.

Euphonic *l’* Whenever clitic doubling occurs, if the verb starts with a vowel it is usually preceded by the particle *l’* (*a lalla a l’ammia o mâ* ‘the aunt looks at the sea’). As it merely plays a euphonic role, we tag it `PART` and attach it to the verb with the relation `dep`.

Language-specific relations We use `expl:pv` for clitics attached to pronominal verbs (*assunnâse* ‘to dream’, *fâghela* ‘to achieve something’), `expl:impers` for the impersonal usage of the pronoun *se* (*in scî cotidien se parla de sti fæti* ‘these facts are being discussed in newspapers’), and `expl:pass` for all uses of *se* as passive marker (*d’autunno se mangia e rostie* ‘roast chestnuts are eaten in autumn’). Similarly to other treebanks, heads of relative clauses are attached to the nominals they modify via `acl:relcl` (Nivre et al., 2016).

4 Corpus statistics

The annotated corpus contains 316 sentences, 6 928 tokens (syntactic words), 1 563 unique surface forms, and 1 192 unique lemmas. Part-of-speech tag and dependency relation statistics for the annotated treebank are shown in Table 2.

In order to get an indication of the quality and consistency of the treebank’s annotations, we test the performance of a standard dependency parser trained on the corpus (Straka and Straková, 2017) using 10-fold cross-validation. The parser, which was trained using the default hyperparameters, achieves 100.0% F1 for tokenisation, 92.00% F1 for lemmatisation, 86.62% F1 for POS tagging, 83.45% F1 for feature prediction, 69.96% UAS and 60.74% LAS. While these scores are not as high as those commonly seen for high-resource languages, they compare favourably to the performance observed for other corpora of similar or even larger sizes (Straka and Straková, 2017; Jónsdóttir and Ingason, 2020, *inter alia*), confirming the consistency of the annotations. An exciting direction for future research would be to explore the possibility of boosting parsing performance via cross-lingual transfer on Italian or Spanish UD data.

Label	Count
det	849
punct	792
case	748
nsubj	416
advmod	412
obl	411
root	316
obj	287
nmod	267
mark	245
cc	243
conj	234
aux	216
amod	150
expl	149
dep	134
expl:pv	125
iobj	105
fixed	103
acl:relcl	102
cop	100
advcl	95
xcomp	95
parataxis	83
ccomp	53
flat	43
acl	35
discourse	23
expl:impers	23
appos	22
nummod	19
dislocated	16
csubj	8
vocative	6
orphan	3

(a) Dependency labels

Tag	Count	Example lemmas
PRON	928	*o che se ghe me*
ADP	904	*de à da in pe*
NOUN	896	*giorno paise parte gio çittæ*
DET	850	*o un quello tutto mæ*
PUNCT	792	*, . ! : ?*
VERB	762	*fâ ëse avei anâ dî*
ADV	459	*no ciù ben tanto ghe*
AUX	318	*ëse avei poei stâ voei*
CCONJ	240	*e ma ò ni comme*
ADJ	219	*bello antigo mæximo santo cao*
PROPN	189	*Zena Gexù Segnô Zane Arbâ*
SCONJ	148	*che se comme perché quande*
PART	136	*l'*
NUM	42	*doî eutto 1929 quaranta quattro*
INTJ	23	*scì ben eh ah no*
X	22	*Tintin aventures de del les*

(b) Part-of-speech tags

Table 2: Corpus annotation statistics

5 Conclusions

We have presented the first corpus of Ligurian annotated according to the Universal Dependencies framework, as well as a set of instructions for the annotation of the less trivial constructions. Additionally, to motivate our choice of linguistic variant and spelling system, we contributed an analysis of the dialects and orthographic standards of Ligurian, setting some guidelines which we hope will prove themselves useful for future contributions of corpora in this language. While the size of this corpus is small compared to the datasets of high-resource Romance languages such as French or Italian, it will now be possible to use this data to bootstrap any future Ligurian annotation efforts.

References

Andrea Acquarone. 2015. Scrivere la lingua. In Andrea Acquarone, editor, *Parlo Ciæo. La lingua della Liguria*, pages 87–94. De Ferrari / Il Secolo XIX, Genoa, Italy.

Graziadio Isaia Ascoli. 1876. Del posto che spetta al ligure nel sistema dei dialetti italiani. *Archivio glottologico italiano*, 2:111–160.

Fabio Canessa. 2016. L'inno naçionâ da Liguria. In Andrea Acquarone, editor, *Riso Ræo, l'antologia de Parlo Ciæo*. De Ferrari.

Arrigo Castellani. 1982. Quanti erano gl'italofoni nel 1861? *Studi linguistici italiani*, 8:3–26.

Tullio De Mauro. 1991. *Storia linguistica dell'Italia unita*. Laterza, Roma-Bari, Italy.

Friedrich Christian Diez. 1836. *Grammatik der romanischen Sprachen*, volume 1.

Werner Forner. 1997. Liguria. In Martin Maiden and Mair Parry, editors, *The dialects of Italy*, pages 198–225. Routledge, London, United Kingdom, and New York, United States of America.

Bruno Guillaume. 2019. Graph Matching for Corpora Exploration. In *JLC 2019 - 10èmes Journées Internationales de la Linguistique de corpus*, Grenoble, France, November.

Johannes Heinecke. 2019. ConlluEditor: a fully graphical editor for Universal dependencies treebank files. In *Universal Dependencies Workshop 2019*, Paris.

Bartolomeo Iacopone. 2017. Derê à scappâ. In *Vuxe de Ligüria*, volume 9. Comune di Pontedassio, Pontedassio, Italy.

Hildur Jónsdóttir and Anton Karl Ingason. 2020. Creating a parallel Icelandic dependency treebank from raw text to Universal Dependencies. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2924–2931, Marseille, France, May. European Language Resources Association.

Stefano Lusito. 2020. Lazarillo de Tormes. In *Cabirda. Lengue e lettiatue romanse*, volume 5. Zona, Genoa, Italy. Translation of an anonymous Spanish novella from 1554.

Stefano Lusito. 2021. Tipologie testuali e modalità di circolazione della prosa contemporanea in genovese. In Giuliano Bernini, Federica Guerini, and Gabriele Iannaccaro, editors, *La presenza dei dialetti italo-romanzi nel paesaggio linguistico: ricerche e riflessioni*, pages 155–174. Bergamo University Press / Sestante Edizioni.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, August. Association for Computational Linguistics.

Fiorenzo Toso. 1995. *Storia linguistica della Liguria*. Le Mani, Recco, Italy.

Fiorenzo Toso. 1997. *Grammatica del genovese*. Le Mani, Recco, Italy.

Fiorenzo Toso. 2001. *Isole tabarchine. Gente, vicende e luoghi di unavventura genovese nel Mediterraneo*. Le Mani, Recco, Italy.

Fiorenzo Toso. 2002. Liguria. In Manlio Cortelazzo, Carla Marcato, and Nicola De Blasi, editors, *I dialetti italiani: storia, struttura, uso*, pages 245–252. UTET, Turin, Italy.

Fiorenzo Toso. 2003a. *Da Monaco a Gibilterra. Storia, lingua e cultura di villaggi e città-stato genovesi verso Occidente*. Le Mani, Recco, Italy.

Fiorenzo Toso. 2003b. *I Tabarchini della Sardegna. Aspetti linguistici ed etnografici di una comunità ligure doltremare*. Le Mani, Recco, Italy.

Fiorenzo Toso. 2009. *La letteratura ligure in genovese nei dialetti locali. Profilo storico e antologia*, volume 1. Le Mani, Recco, Italy.

Fiorenzo Toso. 2018. *A bocca do lô*. De Ferrari, Genoa, Italy. Translation of 1892 novel *La bocca del lupo* by G. Invrea.

Fiorenzo Toso. 2019. O santo evangëio segondo Marco. Forthcoming.

Fiorenzo Toso. 2020. I freschi de gexe do ponente, sensa destin. *O Stafî*, 2(2):4.

Universal Dependencies for Old Turkish

Mehmet Oguz Derin
Morgenrot, Inc.
Turkey
mehmetoguzderin@mehmetoguzderin.com

Takahiro Harada
Morgenrot, Inc.
Advanced Micro Devices, Inc.
USA
takahiro.harada@amd.com

Abstract

We introduce the first treebank for Old Turkic script Old Turkish texts, consisting of 23 sentences from Orkhon corpus and transliterated texts such as poems, annotated according to the Universal Dependencies (UD) guidelines with universal part-of-speech tags and syntactic dependencies. Then, we propose a text processing pipeline for the script that makes the texts easier to encode, input and tokenize. Finally, we present our approach to tokenization and annotation from a cross-lingual perspective by inspecting linguistic constructions compared to other languages.

1 Introduction

Old Turkish[1] (ISO 693-3[2]: otk) was a pluricentric[3] Turkic language with different dialects spoken across Eurasia between the 7th and 14th centuries CE[4], written with different scripts, including Old Turkic script, and its corpora consist of three groups (Ağca, 2021). The modern descendant languages of Old Turkish, a subset of the Turkic language family (Glottocode[5]: comm1245), have more than a hundred million speakers. Some of these languages are classified as endangered by UNESCO[6]. The corpora represent the first sizable record of Turkic languages. Thus, the language and corpora are essential for research into the Turkic language family as they provide clues about stages with scarce data (Savelyev and Robbeets, 2020).

Old Turkic (ISO 15924[7]: Orkh) was a script used to write Old Turkish between the 7th and 10th centuries that reflects characteristics of Turkic languages, including vowel harmony, the binary distinction of non-nasal consonants, letters that sound of the object they depict, and its inventory consists of texts on stelae, papers, and other items including seals, bowls (Erdal, 2004). Materials written with Old Turkic are the very first texts in Old Turkish corpora. The Old Turkic Unicode Range (10C00–10C4F) makes it possible to digitize a subset of the script in a standards-compatible way (The Unicode Consortium, 2021). Such feasibility of maintaining a digital Old Turkic corpus provides an opportunity to compare later Old Turkish corpora texts with prior ones using a unified encoding.

Despite the extensive growth of research and print literature around the Old Turkish language and the Old Turkic script in recent decades, the digitization efforts for Old Turkic script Old Turkish

[1] The language itself still goes by different names in research, we call it by the Old Turkish name to stay consistent with ISO, reserve Old Turkic name for the script, and avoid naming either language or script as Orkhon since it stands better as the name of the corpus. An apparent name clash could be the study of the precedent of the modern Turkish language. However, it almost ubiquitously goes by Old Anatolian Turkish name and not as Old Turkish.

[2] iso639-3.sil.org/code_tables/639/data?title=otk

[3] The rigorous work by Ağca (2021) verifies the observation by Erdal (2004), as "The differences within Old Turkic are by no means greater than, e.g., within Old Greek", that differences could fit in dialectology; hence we briefly call as pluricentric.

[4] For periodization, we adopt the convention by Ağca (2021) since it is a data-based study of texts. The recent work by Johanson (2021) starts the period from the fifth and sixth centuries CE, but we avoid including these centuries where the amount of tangible text is minuscule and extrapolating the grammar as found in Old Turkish corpora texts directly could be misleading.

[5] glottolog.org/resource/languoid/id/comm1245

[6] unesco.org/languages-atlas/, also see endangeredlanguages.com/lang/search/#/?q=Turkic

[7] unicode.org/iso15924/iso15924-codes.html

inscriptions to produce reusable, standards-compatible, open-access computational resources and data are scarce, and advanced tooling for NLP does not exist. Therefore, we have developed the first Universal Dependencies (UD) (de Marneffe et al., 2021) (Nivre et al., 2020) treebank with part-of-speech tags and syntactic annotation for Old Turkic script Old Turkish texts and its tooling to start the NLP applications' building process. We chose the UD scheme because it provides guidelines for consistent annotation of typologically different languages and has extensive adoption and active community, making it possible to validate annotations by specification, data, tools, and discussion. We also built tooling for the treebank to establish a workflow for further research. This small, manually annotated treebank and associated tooling is the first step towards a larger-scale, potentially automated analysis of Old Turkic script encoded Old Turkish texts for UD.

We organized the remainder of this paper as follows. First, in Section 2, we briefly summarize digital or printed related work. Then, in Section 3, we provide an overview of the Old Turkish language and the Old Turkic script. In Section 4, we describe the texts and tools used in building the treebank. In Section 5, we discuss issues with tokenization and sentence segmentation before presenting an approach that makes further annotation consistent, and then, in Section 6, we explain the annotation process for part-of-speech tagging and dependencies in a cross-lingual perspective. Finally, in Section 7, we conclude the paper and contemplate future work.

2 Related Work

Despite the lack of Old Turkish treebanking, there is an increasing body of academic work for treebanking of Turkic languages and studies of the Old Turkish language and the Old Turkic script. Inside the Universal Dependencies project, there are treebanks for Turkish, besides others, by Sulubacak et al. (2016) and Uyghur by Eli et al. (2016) with more than 10K tokens. Despite not living inside the Universal Dependencies project (still, some components of Universal Dependencies take part in the paper), the recent Turkish treebanking approach by Kayadelen et al. (2020) represents a landmark in the consistent annotation that presents guidelines akin to the Short-Unit Word perspective of Japanese treebanking in Universal Dependencies (Omura and Asahara, 2018) (which we use as a convention in our cross-lingual comparisons, besides EWT for English by (Silveira et al., 2014)). Another essential reference for Turkish NLP is the comprehensive work by Oflazer and Saraçlar (2018). Although we mention the Turkish NLP works due to their size compared to other existing languages in the Turkic language family, it is crucial to note that Old Turkish has critical differences from Turkish, not only phonetically but also grammatically. The recent encyclopedic work on the Orkhon corpus by Ercilasun (2016) makes extensive use of literature to provide a methodic reading of the script and the interpretation of the language. The recently published dictionary by Wilkens (2021) provides an essential contribution with its open-access model and focuses on the Old Uyghur corpus. A recent, comprehensive survey of Turkic languages by Johanson (2021) also adopts the open-access model and provides an essential resource for our work. On historical dictionaries that have compilation near Old Turkish period, the renditions made in the last decade on historical Karakhanid bilingual dictionary by Ercilasun and Akkoyunlu (2014), and later historical Old Uyghur by Yunusoğlu (2012), Khwarezmian by Kaçalin and Poppe (2017), and Cuman by Argunşah and Güner (2015) bilingual dictionaries remain as primary references. The grammars by Tekin (1965), Erdal (2004), and Eraslan, Kemal (2012) are comprehensive works with different scopes that help check grammar points. The grammar by Erdal (2004) is especially helpful as it is written in the English language (a feature that eases the correspondence-finding process further when used in tandem with the recent work, which puts the concepts found in the book into a cross-lingual perspective) and includes comparisons that assist with evaluation inside Universal Dependencies context, such as pronominal copula as found in Hebrew. The comparative grammar by Serebrennikov and Gadžieva (2011) provides a bridge between works inside the language family. The textbook treatises by Tekin and Ölmez (2014) and Ölmez (2017) cover a variety of topics in a comprehensively indexing way. The textbook by Ölmez (2017) also includes a word-by-word breakdown of sentences with further morphological analysis, which is the closest work that we can find to anything resembling tagging of sentences. However, by its textbook nature, it does not provide full coverage. For the delimitation of the corpora, the works of Yıldırım (2017), Aydın (2017), Aydın (2018), and Aydın (2019) provide a comprehensive account of Old Turkic script texts, whereas the recent work of Ağca (2021) provides a detailed analysis of Old Turkish corpora's boundaries with special attention on Old Uyghur corpus. For the digitalization of Old Turkic texts, the essential precedents are the often-cited web portal bitig.kz[8] by Abuseitova and Bukhatuly (2005), which does not use the Unicode Old Turkic block to encode the text due to lack of it at the time of its establishment and does not cover the recently found texts, and the atalarmirasi.org[9] by International Turkic Academy (2017) which provides a listing with more brief coverage of their content. An important Turkic language family digitalization work is Chagatai 2.0[10] (Amat et al., 2018), which includes per-sentence annotations with glossing, but it does not cover Old Turkish period.

3 Background

To provide a background for the rest of the paper, in this section, we provide a very brief overview of key features of the Old Turkish language and Old Turkic script.

3.1 Old Turkish Language

As a historical language, Old Turkish belongs to the Turkic family of languages. The three groups of Old Turkish corpora define the language's three main dialects: Orkhon, Old Uyghur, and Karakhanid. Since it represents some of the earliest attestations inside the Turkic language family and to the extent of material the corpora covers, it bears an essential value for studying the languages that are direct descendants of it and the ones branched earlier (such as Chuvash or Sakha), and stands as a bridge for the under-resourced, endangered Turkic languages which preserve archaic features like anticipating numerals (Zhong, 2019). Following are some general characteristics of the Old Turkish language and Turkic languages, which are also present in languages like Japanese and Korean (Han et al., 2020):

1. Dominant word order is subject-object-verb, but rich morphology allows for out-of-order constructions, especially for translated material.
2. Preference for postpositions (suffixing) and verbal endings.
3. Head-final language in which the embedded clause precedes the main clause.

Besides these, the following are some distinguishing features of Old Turkish that separate it within the Turkic language family:

1. Preservation of the /d/ and /ɲ/[11] phonemes inside and at the end of the words.
2. Use of the locative 𐱅𐱃𐰓𐰑𐰀 ta de "at, from"[12] also as an ablative.
3. Presence of 𐰼 er "to be" as a fully conjugated copula.

3.2 Old Turkic Script

As a phonetic script, Old Turkic consists of more than 40 characters, counting variants. Most of these characters represent a single phoneme, and except for five characters, they hint about the backness of vowels between consonants, while some denote a specific consonant cluster or a specific consonant

[8] bitig.kz

[9] atalarmirasi.org/en

[10] uyghur.ittc.ku.edu

[11] When we write phonetic values between forward slashes, we use International Phonetic Alphabet (IPA) per International Phonetic Association et al. (1999) to denote the value instead of the custom phonetic schemes of our reference work.

[12] We use this notation of specialized original-form transliteration "translation" in the rest of the paper. For original-form, the direction is right-to-left, so in this instance, values of original-form would translate to 𐱅𐰀, 𐱃𐰀, 𐰓𐰀, and 𐰑𐰀. For transliteration, the direction is left-to-right, so in this instance, values of transliteration would translate to te, ta, de, and da, corresponding to the order of values in the original-form. Number of readings is combination of all options at all positions.

with a specific vowel. Although there are five characters for spelling vowels, texts do not write open unrounded vowels explicitly unless they are at the end of the word, and some instances omit vowels in non-initial syllables between consonants, reflecting the tongue root harmony of Turkic languages. An essential phenomenon in Old Turkic script texts is the representation of 𐰣 n /n/ followed by 𐰏 g /g/ or 𐰚 k /k/ by single 𐰭 q /ŋ/ at instances, often found when words that end with an alveolar nasal consonant have the dative case. The punctuation is mostly a colon separating words or word groups, sometimes also found as a dot with single color or a colon or dot with colored marking. Whitespace and line breaks also do not bear a meaning most of the time. The Unicode block for Old Turkic script does not include all characters and variants, thus making it infeasible to do one-to-one digitalization of the Old Turkic script corpus. However, it does include enough characters to represent non-included characters phonetically in a way that would allow for direct representation of all characters through conversion when the block expands. The dominant writing direction is right-to-left, but the layout varies between texts, and the writing material varies between different surfaces like paper, stone, mirrors. We provide tables (see Table 1) for vowels and consonants of our digital rendition of the script and transliteration[13]. Following are some essential characteristics of the Old Turkic script:

1. Open rounded vowels are implicit and not written, except for when they are final.
2. Some consonants have synharmonic variants that govern the realization of vowels.
3. Punctuation is very minimal, and its usage is sparse.

4 Corpus

In this section, we present current texts and explain our Old Turkic script and transliterated text encoding. Additionally, we introduce tools that ease the development process and help validate string conformance to our guidelines.

4.1 Texts

Our current Old Turkish treebank consists of 23 manually annotated sentences through treebank-specific tools. Twenty-one of these sentences come from the first face of the first stele of Tuɲukuk inscriptions, a personal account of events the author witness and partake, themselves being Old Turkic script at the source itself but encoded through our pipeline's character mapping scheme. The remaining two of these sentences come from two recently found as syntactically-analogous by Kurnaz (2009) (which resolved the questions about the latter sentence's ambiguous use of particle) poetic sentences to represent a marginal exemplary case of transliteration from later text into Old Turkic script. Our treebank currently contains 341 tokens, with 14.826 tokens per sentence on average.

4.2 Tools

When working with the Old Turkic script block of Unicode directly, there are not many tools other than tools suited for general text or Unicode purposes, and this lack is even present for problems like missing characters in the block. To not give up on directly encoding using Old Turkic script Unicode block, and to take advantage of the fact that a subset of the range can represent all of the vowels and consonants found in Old Turkish corpora (excluding foreign words in non-Old Turkic script texts), we first define a normalization of digital Old Turkic texts and develop a tool to facilitate automatic normalization.

The normalization does two transformations: reduce characters with Orkhon and Yenisei variants to Orkhon only and break up syllabic characters into multiple characters. We base interpretation of syllabic characters on the work of Ercilasun (2016), which presents a consistent, regular approach. We base the second transformation on corresponding instances found in Old Turkic script themselves, and our approach is available in the source code where we store these transformation rules in a JSON file and apply them through a Python script. The normalization also disallows characters other than

[13] Our reference works do not share a common transliteration scheme, and we do not include them due to the space constraints, only presenting ours, which fits the criteria of being representable in lowercase, ASCII only setting.

	Unround		Round	
	Back	Front	Back	Front
Open	𐰀 a	𐰀 e	𐰆 o	𐰇 u
Closed	𐰃 w	𐰃 i	𐰆 o	𐰇 u

Occlusive				Fricative				Nasal		Vibrant		Approximant	
Voiceless		Voiced		Voiceless		Voiced		Voiced		Voiced		Voiced	
Back	Front	Back	Front	Back	Front	Back	Front	Back	Front	Back	Front	Back	Front
𐰴 ak	𐰚 ek	𐰍 ag	𐰏 eg					𐰭 aq	𐰮 eq				
𐰲 ac	𐰲 ec			𐰿 ax	𐱁 ex			𐰪 aj	𐰪 ej			𐰖 ay	𐰘 ey
𐱃 at	𐱅 et	𐰑 ad	𐰓 ed	𐰽 as	𐰾 es	𐰔 az	𐰔 ez	𐰣 an	𐰤 en	𐰺 ar	𐰼 er	𐰞 al	𐰠 el
𐰯 ap	𐰯 ep	𐰉 ab	𐰋 eb			𐰉 av	𐰋 ev	𐰢 am	𐰢 em				

Table 1: Vowels and consonants in Old Turkic script on left followed by our transliteration (for consonants, preceding vowels are not included in transliterations of text, these are for backness notation in a context-free setting) on right. Phonetic regions (of which we omit the row labels for consonants due to space constraints) are figurative, as we serve the table only to facilitate understanding of our transliteration scheme, and we repeat characters that can represent multiple sounds in respective cells, see mentioned works on Old Turkish grammar and Turkic languages in Related Work section for more details about the realization of the sounds.

allowed explicitly like colons to avoid characters introduced by various tools such as right-to-left or left-to-right or redundant line feed or return markers to be part of the treebank through a sanitizer. The normalization results in 33 letters, excluding specifically allowed punctuation. There are 4 vowels, which we spell twice to form digraphs representing closer versions of these vowels consistently if desired, 5 neutral consonants, 24 synharmonic consonants, representing 12 consonants with varying influence on the realization of vowels. We were able to reduce development overhead by adopting this normalization scheme. Thereby, content encoded with Old Turkic in this paper assumes the normalization applied, and they do not graphically cover characters that are either out of our normalization range or not even in the Unicode block.

For the generation of text identifiers and storage in places where only alphabetic lowercase ASCII (potentially with underscore or dash) is allowed, we developed a simple, rule-based bidirectional transliteration scheme that can represent all consonants and vowels alongside currently present punctuation. We also developed a reverse transliterator from ASCII to Old Turkic script that is more permissive to be compatible with manual transliterations that read better. To store the consonants in the lookup table, we precede the ASCII consonant with a closed vowel at the start, and if the consonant is front, we make the preceding vowel an always front vowel, and if the consonant is back, we make it a back vowel, if neither, we make it both front and back vowel, while single space always represents a backness neutralizer, to provide the users with the backness information. We use this transliteration table and usage of the Old Turkic letter that we do not preserve with the second transformation for choosing the backness of consonant as control key to deduce a Keyman[14] keyboard project that allows us to input Old Turkic script in the normalized form across many devices to develop the treebank and surrounding material.

As we lack an automated tokenization module, we store manually annotated token ranges and annotations in a JSON file and use a Python script to extract tokenized and annotated CONLL-U files from our CSV texts, which need a text column present. However, we do not restrict the presence of other columns that might use an extended range of Unicode blocks or define their features which could allow for easier identification of inscriptions with similar names through embedding GeoJSON of the location of the inscription in a column or other means. Furthermore, in the future, we intend to check for duplicate

[14] keyman.com/14

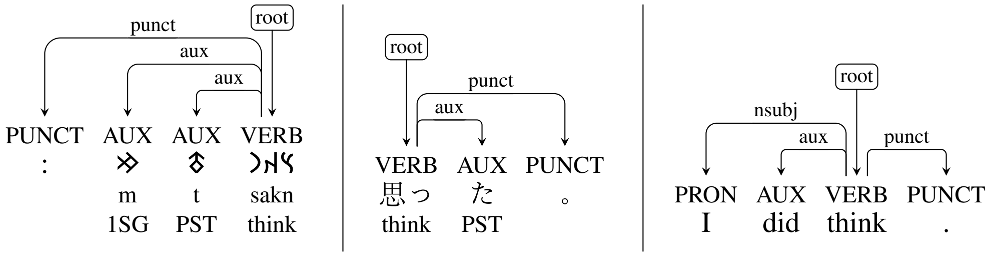


Figure 1: Tokenization and annotation of "I did think." sentence in Old Turkish with comparison to the annotation of figurative translations in Japanese and English. This example highlights the annotation of auxiliaries, especially the person marker, which derives from possessive markers.

sentences before storing them into the CONLL-U file as some inscriptions share verbatim sentences to avoid duplicate content inside the treebank, of which we currently have none.

We also have minor tools that depend on UDAPI products (Popel et al., 2017), such as denoting font automatically for exported TikZ graphs to ease authoring and experimental Anki[15] deck generator from features found in the treebank to demonstrate an edge-case utilization of the Universal Dependencies scheme. We distribute these tools in the not-to-release folder of our treebank.

5 Tokenization and Sentence Segmentation

Tokenization and sentence segmentation of Old Turkic script Old Turkish texts is a challenging task. The script lacks regular punctuation or whitespace for splitting into tokens and a marker for splitting into sentences. Another aspect that makes tokenization harder is letters representing multiple phonemes, but our text processing pipeline eliminates this issue in the resulting output.

5.1 Tokenization

Tokenization requires context-dependent decisions with Old Turkic script Old Turkish texts. Line breaks do not act as tokenizers, especially in limited-space texts, sometimes splitting even the base morpheme. Thus we ignore them in the process of tokenization. Character flipping due to synharmonism also does not act as a consistent tokenizer, and existing treebanking approaches for other Turkic languages also ignore it (for example, they always tokenize question particle despite its second vowel acting according to the harmony). Commonly found colon (or dot in some cases) does meaningful splits, sometimes into words and adpositions, into words, into phrases containing more than one word in other times (and not always consistently, e.g., separating an adjective and a proper noun in some cases while not in other cases). Thus, we always delimit tokens by this punctuation class before further splits. Generalization of such delimitations leads us to treat primarily inflectional morphemes such as possessive markers, case markers, auxiliaries, converbs, tense-aspect-modality-evidentiality (TAME) markers (including personality markers that derive from possessive markers) as tokens to preserve consistency across all of the Old Turkic script texts. We also tokenize particles outlined in the Universal Dependencies guidelines, such as the question particle and other similarly behaving particles in the Old Turkish language, including negation and intensifier particles. If bound morphemes act as nominalizers, resulting in a word that we treat as either noun or pronoun in the Universal Dependencies analysis, we do not split them into tokens and treat them as a single word. We do not treat verbalizer morphemes that impact voice or produce commonly lexicalized verbs while not violating previous steps as individual tokens. This direction results in an approach that provides a rich syntactical analysis and is similar to the recent Turkish treebanking work by Kayadelen et al. (2020) and some Universal Dependencies works like the Japanese language with Short Unit Words perspective (Omura and Asahara, 2018), also a recent highlight in cross-lingual perspective by de Marneffe et al. (2021), and the Shipibo-Konibo language (Vasquez et al., 2018). It is important to note that our guidelines only match with recent work by Kayadelen et al. (2020), and our treebank is the first to adopt this approach in Universal Dependencies Turkic family treebanks.

[15] apps.ankiweb.net

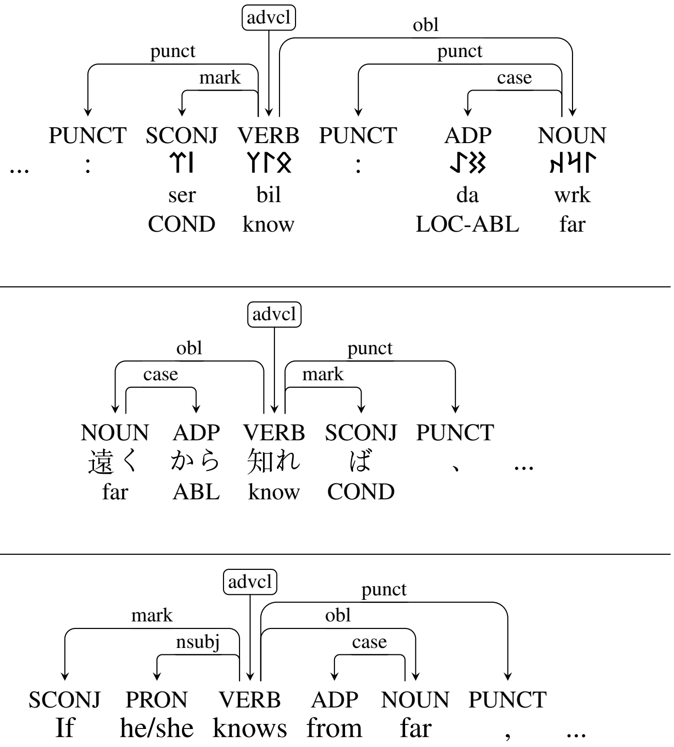


Figure 2: Tokenization and annotation of "If he/she knows from far.." clause in Old Turkish with comparison to annotation of figurative translations in Japanese and English. This example highlights the annotation of conditionals.

Our tokenization guidelines produce entries with characteristics that map into Universal Dependencies guidelines for both tags and dependencies, sitting at balance for cross-lingual perspective inside UD.

5.2 Sentence Segmentation

Sentence segmentation has to be done per the interpretation of the text due to the lack of any punctuation for this matter in Old Turkic script, and not even line breaks act as a regular means of sentence segmentation. After tokenization, we first work through detecting clauses, conjunctions, and finally roots of sentences to do sentence segmentation. We avoid producing parataxis constructions unless found in reported speech, favoring treatment as conjunctions if not fit for sentence split. Our sentence segmentation guidelines produce a set of delimiters that are the union of proposed sentence delimiters in the referenced work for the analyzed text.

6 Annotation

In this section, we go over our application of the Universal Dependencies for annotating parts of speech and syntactic dependencies in a cross-lingual perspective. Currently, our treebank does not have lemma or morphological annotations, and as such, we do not present any guidelines for them, and we only utilize miscellaneous for SpaceAfter=No annotation to all tokens since Old Turkic script texts, as far as we cover, do not contain spaces as a means of separating tokens.

6.1 Part-of-Speech Tagging

We adopt Universal Part-of-Speech (UPOS) tagset as the only convention in our treebank. After tokenization, challenging ones are the bound morphemes and pronominal copulas. We tag possessive (or person) markers as determiners (DET) if they are bound to a noun, but if they act as the only pronominal component of a phrase in a head-final position, we tag them as pronouns (PRON). We tag case markers as adpositions (ADP). We tag verbal endings or converbs that make adverbial clauses subordinating conjunction (SCONJ) or coordinating conjunction (CCONJ) depending on their function.

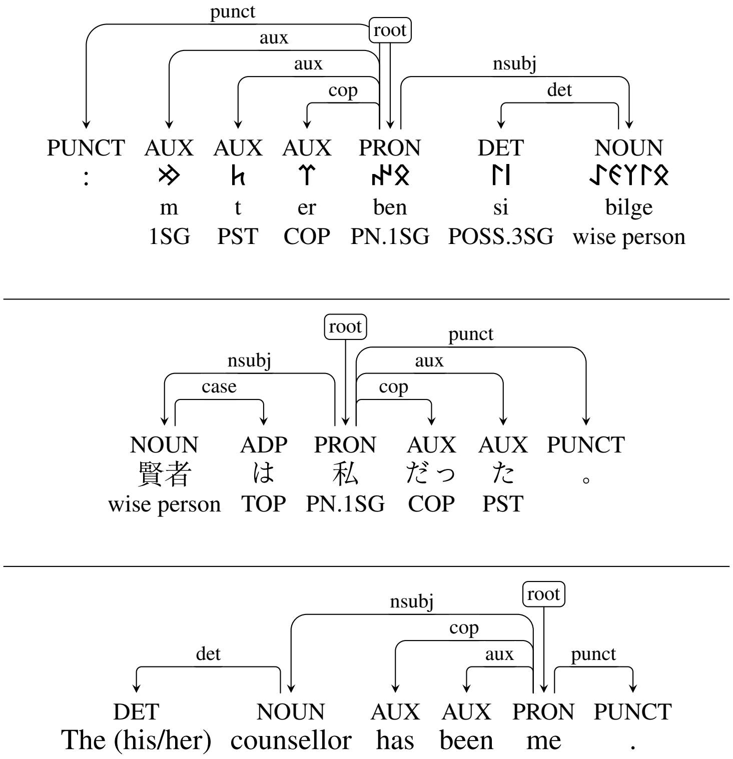

punct
root
aux
aux
nsubj
cop
det
PUNCT AUX AUX AUX PRON DET NOUN
:
m t er ben si bilge
1SG PST COP PN.1SG POSS.3SG wise person
root
punct
nsubj
aux
case
cop
NOUN ADP PRON AUX AUX PUNCT
賢者 は 私 だっ た 。
wise person TOP PN.1SG COP PST
root
nsubj
cop
det
aux
punct
DET NOUN AUX AUX PRON PUNCT
The (his/her) counsellor has been me .

Figure 3: Tokenization and annotation of "The (his/her) counsellor has been me." sentence in Old Turkish with comparison to annotation of figurative translations in Japanese and English. This example highlights the annotation of possessive markers as determiners and auxiliaries.

We tag possessive marker derivative person markers, TAME markers, converbs that act as auxiliary along with a following auxiliary verb, the copula, and verbs that function as auxiliary as auxiliaries (AUX). Per tokenization, auxiliaries are not joined into a single word but instead kept separate units. We do not tag the verbs other than the fully-conjugated copula as an auxiliary (AUX) if they are the clause's predicate. We tag pronominal copulas found at the end of clauses as determiners (DET) per the recommendation of Universal Dependencies guidelines. We tag the regular punctuation as punctuations (PUNCT). Due to their usage in Old Turkish corpora, we treat the word which means “none, no, not, nothing”, and the word which means “all, yes, is, everything” to be pronouns (PRON) as non-interrogative indefinite collective pronouns, a choice shared by the study of Lithuanian Karaim too (Robbeets and Savelyev, 2020), and also in the more recent study of Turkic languages (Johanson, 2021), or similar to other pronouns as determiners (DET) if they act as pronominal copulas. We always tag numbers as numbers (NUM). The rest of the tags map trivially to UPOS by the reference works we use. The treebank currently utilizes 15 tags, leaving out SYM and X. We expect to utilize SYM in the future due to texts containing pictograms. Our tagging approach produces a closed-class for all the tags denoted as such in the Universal Dependencies guidelines.

6.2 Syntactic Annotation

We use universal syntactic relations without subtypes or language-specific relations in our treebank. Out of 37 features, we explicitly avoid using the indirect object (iobj) relation as case markers, such as dative, always follow indirect objects, we use oblique (obl) in such cases, adopting the convention of some Uralic (Partanen and Rueter, 2019) and Japanese (Omura and Asahara, 2018) treebanks for the cross-lingual consistency of annotation. Direct objects (obj) also sometimes have case markers, especially genitive, but we do not treat them as oblique (obl) as they fulfill the core object function. Our treebank currently lacks instances of the clausal subject (csubj) and adnominal clauses (acl) dependencies due to the small data size, and their exact treatment requires special care with head-final characteristic Old Turkish in consideration, bearing challenges similar to Japanese, which we plan to address in future. Out

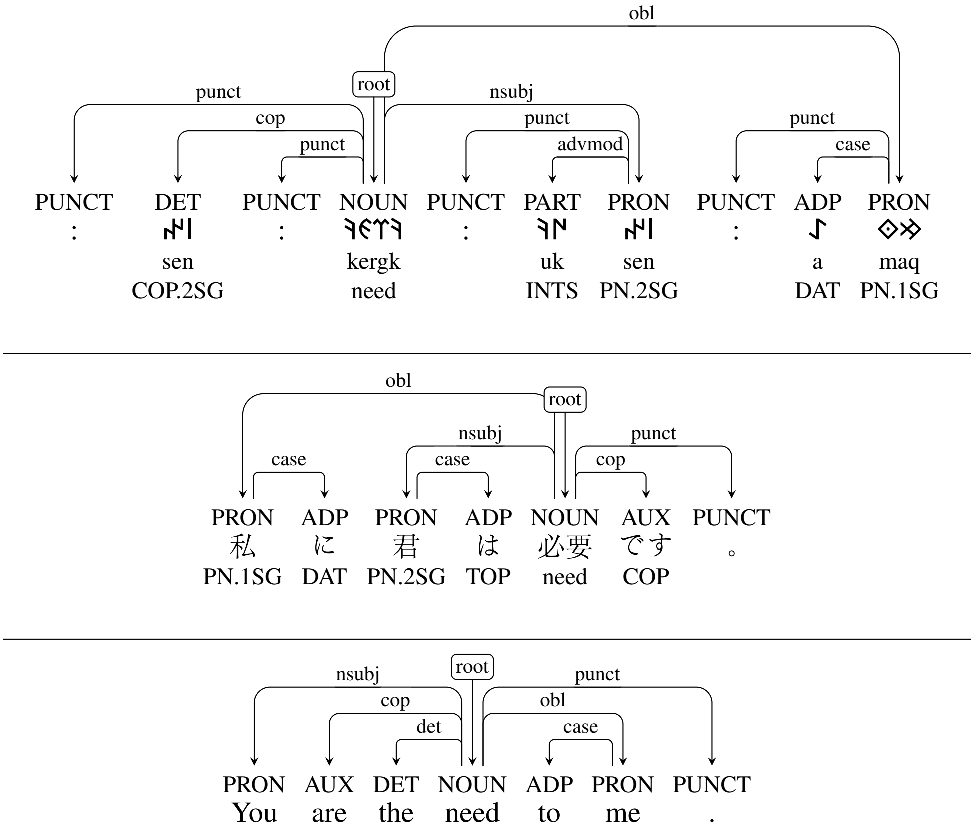

obl
root
punct
nsubj
cop
punct
punct
advmod
punct
case
PUNCT DET PUNCT NOUN PUNCT PART PRON PUNCT ADP PRON
: : : :
sen kergk uk sen a maq
COP.2SG need INTS PN.2SG DAT PN.1SG
obl
root
nsubj
punct
case
case
cop
PRON ADP PRON ADP NOUN AUX PUNCT
私 に 君 は 必要 です 。
PN.1SG DAT PN.2SG TOP need COP
nsubj
root
punct
cop
obl
det
case
PRON AUX DET NOUN ADP PRON PUNCT
You are the need to me .

Figure 4: Tokenization and annotation of "I need you." sentence in Old Turkish with comparison to the annotation of figurative translations in Japanese and English. This example highlights out-of-order construction and pronominal copula.

of other currently unused relations, namely the vocative (vocative), the expletive (expl), the dislocated (dislocated), the classifier (clf), the fixed (fixed), the orphan (orphan), the goes with (goeswith), the reparandum (reparandum), the unspecified dependency (dep) dependencies, only expletive, classifier, and unspecified dependency are unlikely to be utilized in future. We annotate multi-word proper nouns using flat dependency. We annotate question and intensifier particles as adverbial modifiers (advmod). We annotate determiner (DET) tagged pronominal copulas with the copula (cop) relation. If not in proper clausal complement position, we treat reported speech and postposed, non-doubling, parenthetical elements (if we can not annotate as dislocated or appositional) as parataxis. As coordinating conjunction words can sometimes be present at the end of the sequence, we attach them to the element before as coordinating conjunction (cc), which provides a consistent annotation with analogous constructions like phrases formed with antonymy and parallelism markers. If a clausal complement has a null-subject, we annotate the dependency as a clausal complement (ccomp) rather than an open clausal complement (xcomp). We treat punctuations (punct) in line with guidelines while avoiding introducing non-projectivity. Treatment of punctuations might require improvement when treebank size grows as that combined with Universal Dependencies analysis can help further our understanding of punctuation in Old Turkic script texts. We annotate interjections as discourse. Some verbs like “to become, to have” can, depending on their usage, have either an object or a clausal complement attached to them, and we avoid annotating these as copula (cop), reserving the use of relation to the fully conjugated and pronominal copulas. Our tokenization and tagging choices lead to a consistent annotation of dependencies that allows for cross-lingual study.

7 Conclusion and Future Work

Using a cross-lingual perspective, in this paper, we presented the first application of Universal Dependencies to the Old Turkish language with Old Turkic script encoding. The characteristics of the Old Turkish language and the lack of tooling for both the language and the script pose significant challenges. However, as hinted by the extending body of traditional work for the language and recent work in NLP, we have argued through tokenization that it is crucial to define the word concept that creates analogies with other languages. Afterward, we have shown that we developed tooling and guidelines that allow for consistent tokenization, segmentation, tagging, and dependency annotation of the Old Turkish corpora through a finer-grained word definition. The treebank is currently, by its size, insufficient to cover all dependency types in Universal Dependencies or to train a pipeline (Straka et al., 2016) (Honnibal et al., 2020) (Qi et al., 2020), and the tooling does not live under a unified software package but as distinct modules, but it represents an important step towards the enlargement of both the encoded and the annotated text.

In the future, we plan to extend the data size, where we might prioritize using sentences matching recent work that study Old Turkish and its contemporaries in a comparative setting (Kasai, 2014) (Robbeets and Savelyev, 2020) (Lim, 2021) besides extending coverage over the oldest texts in the corpora. We also plan to add lemmas and features, which are crucial for automation due to their governance of how phrases act in a sentence and build additional tooling. As we provide tools for input, character normalization, transliteration, further work should encompass both improvements and extension towards tooling for more accessible span-based annotation of texts potentially through an extension of productive tools for Universal Dependencies (Tyers et al., 2017), automatic tokenization, sentence segmentation, lemmatization, part-of-speech tagging, dependency parsing, and coarser-grained normalization. Another critical area for future work is beginner-friendly guides and materials like dictionaries with references to cross-linguistic colexifications (Rzymski et al., 2020) for providing additional context to interpretations, encouraging people with a less technical background, and also for providing better visibility to the Universal Dependencies community, and if possible, creating avenues for bridging the disconnect in the study of Old Turkish between traditional (often restrained to the language of the work, less accessible towards non-speakers, and not always open-access or in a digitally accessible format) and computational works.

References

M. Abuseitova and B. Bukhatuly. 2005. bitig.kz Turk Bitig.

Ferruh Ağca. 2021. *Dillik Ölçütlere Göre Eski Uygurca Metinlerin Tarihlendirilmesi*. Türk Dil Kurumu Yayınları.

Akbar Amat, Arienne Dwyer, Gülnar Eziz, Alexandre Papas, and CM Sperberg-McQueen. 2018. Annotated Turki Manuscripts from the Jarring Collection Online.

Mustafa Argunşah and Galip Güner. 2015. *Codex Cumanicus*. Kesit Yayınları.

Erhan Aydın. 2017. *Orhon Yazıtları*. Bilge Kültür Sanat Yayıncılık.

Erhan Aydın. 2018. *Uygur Yazıtları*. Bilge Kültür Sanat Yayıncılık.

Erhan Aydın. 2019. *Sibirya'da Türk İzleri*. Kronik Yayıncılık.

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

Marhaba Eli, Weinila Mushajiang, Tuergen Yibulayin, Kahaerjiang Abiderexiti, and Yan Liu. 2016. Universal Dependencies for Uyghur. In *Proceedings of the Third International Workshop on Worldwide Language Service Infrastructure and Second Workshop on Open Infrastructures and Analysis Frameworks for Human Language Technologies (WLSI/OIAF4HLT2016)*, pages 44–50, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Eraslan, Kemal. 2012. *Eski Uygur Türkçesi Grameri*. Türk Dil Kurumu Yayınları.

Ahmet Bican Ercilasun and Ziyat Akkoyunlu. 2014. *Kâşgarlı Mahmud Dîvânu Lugâti' t-Türk*. Türk Dil Kurumu Yayınları.

Ahmet Bican Ercilasun. 2016. *Türk Kağanlığı ve Türk Bengü Taşları*. Derĝah Yayınları.

Marcel Erdal. 2004. *A Grammar of Old Turkic*. Brill, Leiden, The Netherlands.

Ji Yoon Han, Tae Hwan Oh, Lee Jin, and Hansaem Kim. 2020. Annotation issues in Universal Dependencies for Korean and Japanese. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 99–108, Barcelona, Spain (Online), December. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

International Phonetic Association, International Phonetic Association Staff, et al. 1999. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

International Turkic Academy. 2017. Heritage of the Ancestors: Multimedia Fund.

Lars Johanson. 2021. *Turkic*. Cambridge Language Surveys. Cambridge University Press.

Mustafa S. Kaçalin and Nicholas Poppe. 2017. *Mukaddimetü'l-Edeb, Moğolca-Çağatayca Çevirinin Sözlüğü*. Türk Dil Kurumu Yayınları.

Yukiyo Kasai. 2014. The Chinese Phonetic Transcriptions of Old Turkish Words in the Chinese Sources from 6th-9th Century: Focused on the Original Word Transcribed as Tujue. 29:57–135.

Tolga Kayadelen, Adnan Ozturel, and Bernd Bohnet. 2020. A gold standard dependency treebank for Turkish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5156–5163, Marseille, France, May. European Language Resources Association.

Cemal Kurnaz. 2009. Bana Seni Gerek Seni. *Atatürk Üniversitesi Türkiyat Araştırmaları Enstitüsü Dergisi*, 15(39):147–160.

An-King Lim. 2021. On Sino-Turkic verbal functional expressions. *International Journal of Chinese Linguistics*, 8(1):102–138.

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 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May. European Language Resources Association.

Kemal Oflazer and Murat Saraçlar. 2018. *Turkish Natural Language Processing*. Springer.

Mehmet Ölmez. 2017. *Köktürkçe ve Eski Uygurca Dersleri*. Kesit Yayınları.

Mai Omura and Masayuki Asahara. 2018. UD-Japanese BCCWJ: Universal Dependencies Annotation for the Balanced Corpus of Contemporary Written Japanese. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 117–125, Brussels, Belgium, November. Association for Computational Linguistics.

Niko Partanen and Jack Rueter. 2019. Survey of Uralic Universal Dependencies Development. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 78–86, Paris, France, August. 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, May. Association for Computational Linguistics.

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*, pages 101–108, Online, July. Association for Computational Linguistics.

Martine Robbeets and Alexander Savelyev. 2020. *The Oxford Guide to the Transeurasian Languages*. Oxford University Press.

Christoph Rzymski, Tiago Tresoldi, Simon J. Greenhill, Mei-Shin Wu, Nathanael E. Schweikhard, Maria Koptjevskaja-Tamm, Volker Gast, Timotheus A. Bodt, Abbie Hantgan, Gereon A. Kaiping, Sophie Chang, Yunfan Lai, Natalia Morozova, Heini Arjava, Nataliia Hübler, Ezequiel Koile, Steve Pepper, Mariann Proos, Briana Van Epps, Ingrid Blanco, Carolin Hundt, Sergei Monakhov, Kristina Pianykh, Sallona Ramesh, Russell D. Gray, Robert Forkel, and Johann-Mattis List. 2020. The Database of Cross-Linguistic Colexifications, reproducible analysis of cross-linguistic polysemies. *Scientific Data*, 7(1):13, Jan.

Alexander Savelyev and Martine Robbeets. 2020. Bayesian phylolinguistics infers the internal structure and the time-depth of the Turkic language family. *Journal of Language Evolution*, 5(1):39–53, 02.

Boris A Serebrennikov and Ninel Z Gadžieva. 2011. *Türk Yazı Dillerinin Karşılaştırmalı-Tarihî Grameri*. Türk Dil Kurumu Yayınları.

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 Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia, May. European Language Resources Association (ELRA).

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

Talat Tekin and Mehmet Ölmez. 2014. *Türk Dilleri*. BilgeSu Yayıncılık.

Talat Tekin. 1965. *A Grammar of Orkhon Turkic*. University of California, Los Angeles.

The Unicode Consortium. 2021. *The Unicode Standard, Version 14.0.0*. Mountain View, CA. ISBN: 978-1-936213-29-0.

Francis M. Tyers, Mariya Sheyanova, and Jonathan North Washington. 2017. UD Annotatrix: An annotation tool for Universal Dependencies. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 10–17, Prague, Czech Republic.

Alonso Vasquez, Renzo Ego Aguirre, Candy Angulo, John Miller, Claudia Villanueva, Željko Agić, Roberto Zariquiey, and Arturo Oncevay. 2018. Toward Universal Dependencies for Shipibo-konibo. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 151–161, Brussels, Belgium, November. Association for Computational Linguistics.

Jens Wilkens. 2021. *Handwörterbuch des Altuigurischen*. Universitätsverlag Göttingen, Göttingen.

Fikret Yıldırım. 2017. *Irk Bitig ve Orhon Yazılı Metinlerin Dili*. Türk Dil Kurumu Yayınları.

Mağfiret Kemal Yunusoğlu. 2012. *Uygurca-Çince İdikut Sözlüğü*. Türk Dil Kurumu Yayınları.

Yarjis Xueqing Zhong. 2019. *Rescuing a Language from Extinction: Documentation and Practical Steps for the Revitalisation of (Western) Yugur*. Ph.D. thesis, School of Culture History and Language, College of Asia and the Pacific, The Australian National University.

Word Delimitation Issues in UD Japanese

Mai Omura
NINJAL, Japan

Aya Wakasa
NINJAL, Japan

Masayuki Asahara
NINJAL, Japan

{mai-om, awakasa, masayu-a}@ninjal.ac.jp

Abstract

This article discusses word delimitation issues in Universal Dependencies (UD) Japanese. The Japanese language is morphologically rich and does not use white space to delimit words. Word delimitation is an important issue in the development of language resources. Even though UD defines the base unit word using *syntactic words*, UD Japanese utilises **Short Unit Words (SUW)**, which are nearly the same as *morphemes*, the base unit word. We developed another word delimitation version of UD Japanese resources that uses **Long Unit Words (LUW)** as the base unit word, which can be regarded as *syntactic words* in Japanese. We then evaluated their reproducibility through publicly available language resources. The results show that the word delimitation and dependency structure of LUW-based UD Japanese reproduce the results using SUW-based UD Japanese. However, the lemmatisation of LUW is still more complex than that of SUW for a morphologically rich language.

1 Introduction

Universal Dependencies (Nivre et al., 2016) (UD) define the base unit word of dependency annotation as *syntactic words*. Languages with white spaces in their word delimitation tend to utilise space as word boundaries. However, languages that do not use white space to delimit words (e.g. Chinese (Xia, 2000; Leung et al., 2016) and Korean (Chun et al., 2018)) present issues in defining their *syntactic words*. For example, while UD Chinese defines word delimitation using the available word-segmented corpus, UD Classical Chinese (Yasuoka, 2019) did not define syntactic words and utilised characters as the word unit. Even when we use characters as the base unit, the lexicon size is approximately 7,000 for simplified Chinese characters and 13,000 for traditional Chinese characters.

Murawaki (2019) pointed out that the preceding versions of the UD Japanese utilise morphemes as the base unit. The word delimitation is based on the **Short Unit Word** (短単位: hereafter **SUW**), defined by the National Institute for Japanese Language and Linguistics, Japan (hereafter NINJAL). Currently, we have the SUW-based word lexicon UniDic (Den et al., 2007) with 879,222 entries and morpheme-based word embeddings NWJC2vec (Asahara, 2018) with 1,589,634 entries for Japanese. The large size of the lexicon is because Japanese is a morphologically rich language. When we use a longer word unit as the base unit, the lexicon size is larger, and the token type ratio becomes larger. Practically, SUW-based UD Japanese resources can be developed with less effort from publicly available language resources. Thus, we had utilised SUW as the base unit word in UD Japanese.

We newly developed another version of UD Japanese with **Long Unit Word** (長単位: hereafter **LUW**) delimitation. The LUW definition by NINJAL can be regarded as *syntactic words* in Japanese. Even though LUW delimitation is appropriate for the base unit words of UD, the cost of LUW-based corpus development is much higher than that of SUW-based corpus development. Furthermore, the reproducibility of LUW-based UD Japanese should be investigated.

This paper presents LUW-based UD Japanese language resources. We also present the reproducibility of LUW-based UD Japanese structures using currently available tools and language resources. The remainder of this paper is organised as follows. Section 2 presents word delimitation in Japanese, including currently available tools and language resources. Section 3 presents LUW-based UD Japanese language resources. Sections 4 and 5 present an experimental evaluation of their reproducibility. Finally, Section 6 concludes the paper.

2 Word Delimitation in Japanese

2.1 Japanese word delimitation standards by NINJAL

Min. Unit	‖ 全 ‖ 学 ‖ 年 ‖ に ‖ わたっ ‖ て ‖ 小 ‖ 学 ‖ 校 ‖ の ‖ 国 ‖ 語 ‖ の ‖ 教 ‖ 科 ‖ 書 ‖ に ‖ 大 ‖ 量 ‖ の ‖ 挿し ‖ 絵 ‖ が ‖ 用い ‖ られ ‖ て ‖ いる ‖
SUW	‖ 全 ‖ 学年 ‖ に ‖ わたっ ‖ て ‖ 小 ‖ 学校 ‖ の ‖ 国語 ‖ の ‖ 教科 ‖ 書 ‖ に ‖ 大量 ‖ の ‖ 挿し絵 ‖ が ‖ 用い ‖ られ ‖ て ‖ いる ‖
LUW	‖ 全学年 ‖ にわたって ‖ 小学校 ‖ の ‖ 国語 ‖ の ‖ 教科書 ‖ に ‖ 大量 ‖ の ‖ 挿し絵 ‖ が ‖ 用い ‖ られ ‖ ている ‖
Bunsetsu	‖ 全学年にわたって ‖ 小学校の ‖ 国語の ‖ 教科書に ‖ 大量の ‖ 挿し絵が ‖ 用いられている ‖
(*romanisation*)	‖ zen gakunen ni watatte ‖ syou gakkou no ‖ kokugo no ‖ kyoukasho ni ‖ tairyou no ‖ sashie ga ‖ mochii rare te iru ‖
(*gloss*)	‖ for all school years ‖ elementary school-GEN ‖ Japanese language-GEN ‖ textbooks-DAT ‖ many ‖ picture-PL-SBJ ‖ use-PASS-PRET ‖

Translation: *Many pictures are used in elementary school textbooks for all school years.*

Figure 1: Example of Minimum Unit, SUW, LUW, and Bunsetsu in BCCWJ PB33_00032

NINJAL defines several word delimitation standards: Minimum Unit (最小単位), SUW, LUW, and *Bunsetsu* (文節), shown in Figure 1 (Den et al., 2008).

The Minimum Unit standard (最小単位) is defined by word types. Japanese has the following word types: Chinese-origin words (漢語), Japanese-origin words (和語), Loan words other than Chinese-origin words (外来語), Symbols (記号), Numerals (数値表現), and Proper nouns (固有名詞). Chinese-origin words are split into individual characters. Japanese-origin words are split into their shortest units. Loan words other than Chinese-origin words are split into the original shortest unit. Numerals are split into the pronounceable decimal digits. For example, "1076" is split into "千" (*sen*; one thousand), "七十" (*nanajuu*; seventy) and "六" (*roku*; six). Symbols are split into individual characters. Proper nouns are split into their shortest units.

SUWs are defined by the Minimum Unit standards for their word type: Minimal Unit lexicon MORPH $= \{m_1, \ldots\}$ with word types WORDTYPE $= \{wt_{m_1}, \ldots\}$. SUW is defined as follows: If a word is a Minimal Unit (word $\in$ MORPH), then word is SUW. If a word is split into two Minimal Units m_A, m_B and their word types are the same ($wt_{m_A} = wt_{m_B}$), then word is SUW. Note that if a word is split into more than two Minimal Units, the word is not SUW.

Parts of speech (POSs) can be assigned to SUWs. In Japanese, verbs, adjectives, and auxiliary verbs have conjugations. These three POSs have conjugation types (CTYPEs) and conjugation forms (CFORMs). SUW can be categorised as dependent (付属語) and independent words (自立語) by POS. These two correspond to functional and content words in UD. Postposition (助詞), auxiliary verb (助動詞), prefix (接頭辞) and suffix (接尾辞), are categorised as dependent words. The conjugation types are defined by their conjugation patterns, such as class-5 verbs (五段), class-1 verbs (一段), and irregular verbs. The conjugation forms are defined as irrealis form (未然形), conjunctive form (連用形), and so on.

LUW is defined by the Bunsetsu (文節) delimitation. Before defining the LUW delimitation, we define the Bunsetsu delimitation. Bunsetsu is a base phrase in Japanese, which is similar to *eojeol* (語節) in Korean. Bunsetsu composes one compound independent word and dependent words, such as prefix morphemes, postpositions, and auxiliary verbs. Bunsetsu-based Japanese dependency structures have the following properties useful when developing dependency parsers: They are (a) mostly projective,

(b) strictly head-final, and (c) easily produce Bunsetsu delimitation by chunkers. Bunsetsu-based dependency parsers have mainly developed been in the Japanese natural language processing fields (Kudo and Matsumoto, 2002; Kawahara and Kurohashi, 2006). However, since Bunsetsu is a base phrase, POS is not assigned to the Bunsetsu unit.

LUW delimitation is defined as constituents in the Bunsetsu. Because LUWs have their POS and morphological features of conjugation, we can use LUW as the *syntactic words* in the UD standard. One compound-dependent word with prefix morphemes is the semantic head LUW word in Bunsetsu. Most of the SUWs of postpositions, auxiliary verbs, and suffixes are regarded as one LUW. However, NINJAL LUW delimitation defines multi-word functional expression as one LUW.

2.2 Language resource availability

	Lexicon	Word Segmenter	Segmented Corpus	Word Embeddings	TTR
Characters	UTF-8 charset	buildable	buildable	buildable	0.00004
Minimal Unit	N/A	N/A	N/A	N/A	N/A
SUW	UniDic	MeCab	BCCWJ	NWJC2vec	0.00176
LUW	N/A	Comainu	BCCWJ	N/A	0.02922
Bunsetsu	N/A	Comainu	BCCWJ	N/A	0.22221

Table 1: Language resource availability

This section presents availability of the language resources. Table 1 shows the language resource availability for the delimitation. Characters can be produced by simple scripts. Because the Minimal Unit is the unit to determine SUW delimitation manually, there is no publicly available resource for doing so. UniDic [1] is an SUW-based lexicon which can be used in the word segmenter MeCab (Kudo et al., 2004) [2]. Neither LUW nor Bunsetsu lexicons currently exist. The chunker Comainu [3] (Kozawa et al., 2014) can produce LUW and Bunsetsu based on MeCab outputs. SUW, LUW, and Bunsetsu are annotated in the Balanced Corpus of Contemporary Written Japanese (hereafter BCCWJ) (Maekawa et al., 2014).

The column 'TTR' represents the type-token ratio of the units for the BCCWJ. The TTR of Character is $0.00004 = 7,622/195,898,039$; the TTR of SUW is $0.00176 = 185,136/104,612,418$; the TTR of LUW is $0.02922 = 2,434,721/83,308,386$; and the TTR of Bunsetsu is $0.22221 = 9,485,940/42,688,154$. The large TTR causes modelling difficulty for word embeddings. Therefore, Japanese natural language processing uses word embeddings based on SUW (Asahara, 2018) or characters.

2.3 History of UD Japanese word delimitation

UD Japanese KTC (Tanaka et al., 2016) is the UD corpus based on the Kyoto Corpus. The corpus was resegmented into LUW-like word units and a manually annotated phrase structure tree. The phrase-structure tree was then converted into the UD version 1 standard. However, the maintenance of the UD Japanese KTC stopped after the UD version 2.0 standard.

UD Japanese GSD and PUD are original products by Google (McDonald et al., 2013) and were maintained until version 1.4. The UD Japanese team have maintained them from v2.0 (Tanaka et al., 2016). The word delimitation of v2.0-v2.5 was produced by IBM word segmenter (Kanayama et al., 2000) and manually fixed. Those of v2.6-v2.8 treebanks were based on manual annotation of SUW.

UD Japanese BCCWJ is based on the BCCWJ. As mentioned earlier, the BCCWJ has three delimitations: SUW, LUW, and Bunsetsu. Currently, we only use SUW for word delimitation of the UD Japanese BCCWJ (Omura and Asahara, 2018).

[1] https://ccd.ninjal.ac.jp/unidic/en/

[2] https://taku910.github.io/mecab/

[3] https://github.com/skozawa/Comainu

3 LUW-based UD Japanese

		Sentences	Bunsetsus/LUW	Words/LUW	Bunsetsus/SUW	Words/SUW
BCCWJ	train	40,801	308,648	715,759	308,679	908,738
	dev	8,427	60,697	145,398	60,722	178,306
	test	7,881	56,332	134,475	56,350	166,859
GSD	train	7,050	57,174	130,298	57,357	168,333
	dev	507	4,186	9,531	4203	12,287
	test	543	4,568	10,429	4,588	13,034
PUD	test only	1,000	9,971	22,910	10,008	28,788

Table 2: Basic statistics of the LUW-based word delimitation UD.

We developed an LUW-based UD Japanese corpus based on the UD Japanese BCCWJ, GSD, and PUD. Table 2 shows the basic statistics of SUW, LUW, and Bunsetsu in these treebanks. As can be seen in Table 2, the number of LUW words is small because it contains SUWs. Although LUW delimitation is used to define constituents in the Bunsetsu in the previous section, the number of Bunsetsu also declines because multi-word functional expressions are one LUW.

UD Japanese GSD and PUD are annotated with SUW-based word delimitation, UniDic POS information (XPOS), and Bunsetsu-based dependency relations. Version 2.8 of these treebanks were developed using the conversion rules from the Bunsetsu-based dependency structure, which was originally used in the UD Japanese BCCWJ (Omura and Asahara, 2018).

We manually annotated LUW-based word delimitation, POS, and LEMMA for the UD Japanese GSD and PUD. When we found a discrepancy between SUW and LUW, we modified the SUW-based annotations. The conversion rules adopted both SUW and LUW POS and morphological features. The original data before the conversion are available in the Github repository [4]. Note that the BCCWJ initially has LUW-based word delimitation, POS, and LEMMA information.

4 Experimental Settings

We performed experiments to evaluate the reproducibility of versions 2.5, 2.8, and LUW of the UD Japanese GSD with publicly available language resources: **v2.5 (IBM)** is IBM-word-segmenter-based word delimitation; **v2.8 (SUW)** is SUW word delimitation; **LUW** is LUW word delimitation. The data are split into train, dev, and test. We use train for the training, dev for parameter tuning, and test for evaluation.

The evaluation is performed in three layers for each setting. The first layer is the evaluation of all analysers, whose inputs are raw sentences. The second layer is the evaluation of POS tagging and dependency analysers, whose input is word-delimited sentences (**Gold**). The third layer is the evaluation of dependency analysers, whose input is gold word delimited and POS tagged sentences (**Gold**).

We used UDPipe (Straka and Straková, 2017) as trainable pipeline analysers for tokenisation, tagging, lemmatisation, and dependency parsing. We retrained the UDPipe model with the three-word delimitation of v2.5, v2.8, and LUW corpus (**UDPipe (T)** and **Train**). We used UDpipe v1.2.0 [5]. Since the UDPipe model provided by the LINDAT/CLARIN infrastructure [6] was initially trained by v2.5, we also included the result of the original UDPipe model (**UDPipe (O)** and **Original**). The training of dependency analysis layers of UDPipe can use externally trained word embeddings. We compared the results with and without SUW-based word embeddings, NWJC2vec (Asahara, 2018) (**Train w/o vec** or **Train w/ vec**). SUW and LUW can be reproduced by the morphological analysers MeCab and chunker Comainu, which are trained on data other than UD Japanese. **MeCab** means that we use MeCab-0.996

[4] https://github.com/masayu-a/UD_Japanese-GSDPUD-CaboCha

[5] https://ufal.mff.cuni.cz/udpipe/1

[6] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3131

with UniDic-2.1.2 output for the tokenisation. **Comainu** means that we use Comainu-0.72 output for the tokenisation. [7]

We used evaluation scripts of CoNLL 2018 shared tasks (Zeman et al., 2018). **Words**, **UPOS**, **XPOS**, and **Lemma** are their F_1 scores. **UAS** (Unlabelled Attachment Score) and **LAS** (Labelled Attachment Score) are standard evaluation metrics in dependency parsing results. **CLAS** (Nivre and Fang, 2017) is defined as the labelled F_1-score over all relations except functional and punctuation relations based on **LAS**. **MLAS** (Zeman et al., 2018) is an extension of CLAS, in which function words are not ignored, but treated as features of content words. In addition, the part-of-speech tags and morphological features are evaluated. **BLEX** (Zeman et al., 2018) is another extension of CLAS, incorporating lemmatisation instead of morphological features.

5 Results

Table 3 shows the results. First, we show the reproducibility of word delimitation. Whereas the word delimitation of v2.5 (IBM) was 91.94-91.96% by UDPipe, the values of v2.8 (SUW) and LUW were 96.14% and 95.02%, respectively. SUW and LUW thus are significantly more reproducible than v2.5. When we used MeCab for SUW and Comainu for LUW, the word delimitation accuracies were 96.84% and 97.19%, respectively.

Second, we confirmed the results of UPOS and XPOS. When we used Gold word segmentation as the input, the accuracies of UPOS and XPOS were 96.82-97.39% and 96.34-96.70%, respectively. When we used raw sentences as the input for UDPipe, v2.5 UPOS and XPOS accuracies were 89.3% and 88.98%, respectively, because of their low word-delimitation accuracy. The UPOS and XPOS accuracies of the SUW were 93.96% and 93.29%, respectively. The UPOS and XPOS accuracies of LUW were 92.37% and 92.16%, respectively. When we used MeCab for the tokeniser, the UPOS and XPOS, accuracies of SUW by UD Pipe are 94.42% and 93.58%, respectively. When we used Comainu for the tokeniser, UPOS and XPOS accuracies of LUW by UD Pipe remained at 94.34% and 94.18%, respectively.

Third, the result of lemmatisation shows the disadvantage of LUW. When we used Gold word segmentation as the input, v2.5 and SUW showed 98.93-99.20% LEMMA accuracy. However, LUW showed 93.78%. When we used raw sentences as the input for UDPipe, the LEMMA accuracy of LUW was 89.74%. This is because the lemmatisation of compound morphemes in Japanese is not straightforward. We need other lemmatisation modules for LUW word lemmatisation. When we used Comainu for word delimitation, the LEMMA accuracy of LUW by UD Pipe was 91.32% despite the high tokenisation accuracy (97.19%).

Next, we discuss dependency analysis accuracy. When we used Gold word delimitation, v2.5 outperformed the others. However, because the word delimitation accuracy of v2.5 was low, the UAS and LAS scores dropped from 93.36-95.20% to 75.43-77.91% for the raw sentence. The UAS and LAS scores of SUW were 85.22% and 83.50% with UDPipe, and 88.22% and 86.32% with MeCab for the raw sentences. The UAS and LAS scores of LUW were 83.49% and 82.07% with UDPipe, and 88.16% and 86.45% with Comainu for the raw sentences. When using publicly available word segmenters (MeCab and Comainu), the difference between SUW and LUW for dependency analysis accuracy (UAS, LAS) was not significant. Whereas the CLAS and MLAS results are similar to the UAS and LAS results, the BLEX of LUW is significantly lower than that of SUW. This is because the lemmatisation of LUW is quite difficult. Despite the low lemmatisation accuracy of LUW, the BLEX of LUW outperforms that of v2.5.

[7] These tools can output the XPOS; however, these experiments have ignored inconsistent results.

Treebank	Tokenisation	POS Tagging	Dep. Analysis	Words	UPOS	XPOS	Lemmas	UAS	LAS	CLAS	MLAS	BLEX
v2.5 (IBM)	**UDPipe (O)**	**UDPipe (O)**	**Original**	91.96%	89.35%	88.98%	91.19%	77.91%	76.45%	66.50%	63.94%	65.98%
	Gold	**UDPipe (O)**	**Original**	-	96.93%	96.41%	99.07%	92.70%	90.65%	83.55%	80.64%	82.84%
	Gold	**Gold**	**Original**	-	-	-	-	94.85%	93.77%	87.08%	86.98%	87.08%
	UDPipe (T)	**UDPipe (T)**	**Train w/o vec**	91.94%	89.30%	89.00%	91.14%	77.11%	75.43%	64.84%	62.69%	64.42%
	Gold	**UDPipe (T)**	**Train w/o vec**	-	96.82%	96.34%	98.93%	92.12%	89.82%	82.10%	79.19%	81.37%
	Gold	**Gold**	**Train w/o vec**	-	-	-	-	94.68%	93.36%	86.29%	86.10%	86.29%
	UDPipe (T)	**UDPipe (T)**	**Train w/ vec**	91.94%	89.30%	89.00%	91.14%	77.31%	75.87%	65.71%	63.63%	65.33%
	Gold	**UDPipe (T)**	**Train w/ vec**	-	96.82%	96.34%	98.93%	92.58%	90.58%	83.58%	80.76%	82.84%
	Gold	**Gold**	**Train w/ vec**	-	-	-	-	95.20%	94.15%	87.85%	87.71%	87.85%
v2.8 (SUW)	**MeCab**	**UD Pipe (T)**	**Train w/o vec**	96.84%	94.42%	93.58%	96.07%	87.38%	85.40%	77.57%	75.04%	77.14%
	UDPipe (T)	**UDPipe (T)**	**Train w/o vec**	96.14%	93.96%	93.29%	95.39%	84.40%	82.58%	75.20%	72.81%	74.79%
	Gold	**UDPipe (T)**	**Train w/o vec**	-	97.39%	96.52%	99.20%	91.20%	89.10%	83.10%	80.32%	82.64%
	Gold	**Gold**	**Train w/o vec**	-	-	-	-	92.28%	91.12%	85.54%	85.00%	85.54%
	MeCab	**UDPipe (T)**	**Train w/ vec**	96.84%	94.42%	93.58%	96.07%	88.22%	86.32%	79.28%	76.74%	78.83%
	UDPipe (T)	**UDPipe (T)**	**Train w/ vec**	96.14%	93.96%	93.29%	95.39%	85.22%	83.50%	76.85%	74.48%	76.44%
	Gold	**UDPipe (T)**	**Train w/ vec**	-	97.39%	96.52%	99.20%	92.05%	90.07%	84.90%	82.13%	84.40%
	Gold	**Gold**	**Train w/ vec**	-	-	-	-	93.95%	93.32%	89.07%	88.67%	89.07%
LUW	**Comainu**	**UDPipe (T)**	**Train w/o vec**	97.19%	94.34%	94.18%	91.32%	87.91%	86.16%	78.10%	74.19%	71.78%
	UDPipe (T)	**UDPipe (T)**	**Train w/o vec**	95.02%	92.37%	92.16%	89.74%	83.25%	81.83%	72.31%	68.54%	67.10%
	Gold	**UDPipe (T)**	**Train w/o vec**	-	96.90%	96.70%	93.78%	92.82%	90.93%	82.66%	78.51%	75.66%
	Gold	**Gold**	**Train w/o vec**	-	-	-	-	93.86%	93.23%	85.77%	85.33%	85.77%
	Comainu	**UD Pipe (T)**	**Train w/ vec**	97.19%	94.34%	94.18%	91.32%	88.16%	86.45%	78.69%	75.05%	72.52%
	UDPipe (T)	**UDPipe (T)**	**Train w/ vec**	95.02%	92.37%	92.16%	89.74%	83.49%	82.07%	72.85%	69.15%	67.69%
	Gold	**UDPipe (T)**	**Train w/ vec**	-	96.90%	96.70%	93.78%	93.18%	91.26%	83.27%	79.30%	76.43%
	Gold	**Gold**	**Train w/ vec**	-	-	-	-	94.03%	93.74%	87.19%	86.86%	87.19%

Table 3: Results: Reproducibility of versions 2.5, 2.8, and LUW of UD Japanese GSD

Treebank	**Tokenisation**	**UAS**			**LAS**		
		w/o Vec	w/vec	Diff	w/o Vec	w/vec	Diff
v2.5(IBM)	UDPipe	77.11%	77.31%	+0.20	75.43%	75.87%	+0.44
v2.8(SUW)	UDPipe	84.40%	85.22%	+0.82	82.58%	83.50%	+0.92
v2.8(SUW)	MeCab	87.38%	88.22%	+0.84	85.40%	86.32%	+0.92
LUW	UDPipe	83.25%	83.49%	+0.24	81.83%	82.07%	+0.24
LUW	Comainu	87.91%	88.16%	+0.25	86.16%	86.45%	+0.29

Table 4: Effect of Word Embeddings (Subset of Table 3)

Finally, we confirmed the effect of word embeddings for UDPipe. Table 4 shows the effect of word embeddings. The word embeddings NWJC2vec (Asahara, 2018) is based on SUW. Thus, whereas the dependency accuracy of IBM and LUW increased by 0.20-0.44 and 0.24-0.29, respectively, the dependency accuracy of SUW increased by 0.82-0.92. The results suggest that the availability of word embeddings is another important factor in the development of UD language resources. As shown by the presented token-type ratios in Table 1, LUW-based word embeddings are not practical in the current state of Japanese natural language processing. Even though the SUW word embeddings are a subset of LUW word definitions, the dependency accuracy of LUW is comparable to that of SUW.

6 Conclusions

This article presented word delimitation issues in UD Japanese. We provided an overview of the word delimitation standards and the history of UD Japanese, and then developed LUW-based UD Japanese language resources that adopt the word unit as a *syntactic word* in Japanese. We evaluated the reproducibility of several versions of UD Japanese with publicly available resources. The results show that LUW-based UD Japanese is as reproducible as SUW-based UD Japanese, even though LUW-based word embeddings are not available. Lemmatisation of LUWs is still difficult because of their compound morphological structures.

Annotation of the *syntactic word*-based dependency treebank is a difficult task for morphologically rich languages such as Japanese without word delimitation. It took great effort to define morphemes, POSs, compound word constructions, and dependency structures. The work took more than eight years to complete and was finished in 2021. The data were released as version 2.9 of UD Japanese GSDLUW, UD Japanese PUDLUW, and UD Japanese BCCWJLUW.

Our future work will be to adjust the differences in opinions in Japanese natural language processing communities for word delimitation issues. We are also planning to adjust the difference in word delimitation among East Asian languages, such as Chinese and Korean.

Acknowledgements

This work was supported by JSPS KAKENHI (Grant Number JP17H00917) and is a project of the Center for Corpus Development, NINJAL.

References

Masayuki Asahara. 2018. NWJC2Vec: Word embedding dataset from ‘ NINJAL Web Japanese Corpus ’. *Terminology: International Journal of Theoretical and Applied Issues in Specialized Communication*, 24:7–22, January.

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)*, pages 2194–2202. European Language Resources Association (ELRA), May.

Yasuharu Den, Ogiso Toshinobu, Ogura Hideki, Yamada Atsushi, Minematsu Nobuaki, Uchimoto Kiyotaka, and Koiso Hanae. 2007. The development of an electronic dictionary for morphological analysis and its application to Japanese corpus linguistics [in japanese]. *Japanese Linguistics*, 22:101–123, October.

Yasuharu Den, Junpei Nakamura, Toshinobu Ogiso, and Hideki Ogura. 2008. A proper approach to Japanese morphological analysis: Dictionary, model, and evaluation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 1019–1024. European Language Resources Association (ELRA), May.

Hiroshi Kanayama, Kentaro Torisawa, Yutaka Mitsuishi, and Jun'ichi Tsujii. 2000. A hybrid Japanese parser with hand-crafted grammar and statistics. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*, pages 411—-417, July.

Daisuke Kawahara and Sadao Kurohashi. 2006. A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 176–183. Association for Computational Linguistics, June.

Shunsuke Kozawa, Kiyotaka Uchimoto, and Yasuharu Den. 2014. Adaptation of Long-Unit-Word analysis system to different part-of-speech tagset [in Japanese]. *Journal of Natural Language Processing*, 21(2):379–401.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, pages 1—7. Association for Computational Linguistics, August.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237. Association for Computational Linguistics, July.

Herman Leung, Rafaël Poiret, Tak-sum Wong, Xinying Chen, Kim Gerdes, and John Lee. 2016. Developing Universal Dependencies for Mandarin Chinese. In *Proceedings of the 12th Workshop on Asian Language Resources (ALR12)*, pages 20–29. The COLING 2016 Organizing Committee, December.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguti, Makiro Tanaka, and Yasuharu Den. 2014. Balanced corpus of contemporary written Japanese. *Language resources and evaluation*, 48(2):345–371, December.

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. Association for Computational Linguistics, August.

Yugo Murawaki. 2019. On the definition of Japanese word, June. arXiv: 1906.09719 [cs.CL].

Joakim Nivre and Chiao-Ting Fang. 2017. Universal Dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 86–95. Association for Computational Linguistics, May.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666. European Language Resources Association (ELRA), May.

Mai Omura and Masayuki Asahara. 2018. UD-Japanese BCCWJ: Universal Dependencies annotation for the Balanced Corpus of Contemporary Written Japanese. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 117–125. Association for Computational Linguistics, November.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing ud 2.0 with UDpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99. Association for Computational Linguistics, August.

Takaaki Tanaka, Yusuke Miyao, Masayuki Asahara, Sumire Uematsu, Hiroshi Kanayama, Shinsuke Mori, and Yuji Matsumoto. 2016. Universal dependencies for japanese. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1651–1658. European Language Resources Association (ELRA), May.

Fei Xia. 2000. The segmentation guidelines for the Penn Chinese Treebank (3.0). *University of Pennsylvania*, November.

Koichi Yasuoka. 2019. Universal Dependencies Treebank of the Four Books in Classical Chinese. In *DADH2019: 10th International Conference of Digital Archives and Digital Humanities*, pages 20–28. Digital Archives and Digital Humanities, December.

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. Association for Computational Linguistics, October.

Numerals and what counts

Jack Rueter
Department of Digital Humanities
University of Helsinki
jack.rueter@helsinki.fi

Niko Partanen
Department of Finnish,
Finno-Ugrian and Scandinavian Studies
University of Helsinki
niko.partanen@helsinki.fi

Flammie A. Pirinen
Divvun
Uit Norgga árktalaš universitehta
Tromsø, Norway
tommi.pirinen@uit.no

Abstract

This study discusses the way different numerals and related expressions are currently annotated in the Universal Dependencies project, with a specific focus on the Uralic language family and only occasional references to the other language groups. We analyse different annotation conventions between individual treebanks, and aim to highlight some areas where further development work and systematization could prove beneficial. At the same time, the Universal Dependencies project already offers a wide range of conventions to mark nuanced variation in numerals and counting expressions, and the harmonization of conventions between different languages could be the next step to take. The discussion here makes specific reference to Universal Dependencies version 2.8, and some differences found may already have been harmonized in version 2.9. Regardless of whether this takes place or not, we believe that the study still forms an important documentation of this period in the project.

1 Introduction

Numerous treebanks in the Uralic languages have become available within the *Universal Dependencies* (UD) project (Zeman et al., 2021). In recent years, at least within the Uralic language family, we have seen new treebanks emerging in languages with closely related siblings that already have an existing treebank. Examples of such languages are Skolt Saami, in relation to Northern Saami (Tyers and Sheyanova, 2017), Komi-Permyak, in relation to Komi-Zyrian (Partanen et al., 2018), or Moksha in relation to Erzya (Rueter and Tyers, 2018). Although the entirety of Uralic languages is still not fully represented within the Universal Dependencies project, the situation has improved in many ways since the last survey on the state of this language family in UD was conducted (Partanen and Rueter, 2019). While more extensive surveys are useful, we think there are situations where individual nuanced features should be compared between the languages, so that consistency could be maintained and improved upon. At the same time, this may provide a thoughtful point of departure for new discussions around such features, as we believe the questions discussed here are relevant beyond the realm of Uralic languages. Even in other treatment of UD on different language groups, such as Slavic, numerals have been recognized as one category that demands special attention (Zeman, 2015). Recently Schneider and Zeldes have also discussed inconsistent nominal constructions in the English treebanks (2021), and even the issues we describe in the Uralic treebanks here can well be described in a similar vein. These are not dramatic issues, but small points of divergence that we could pay attention to, but if we decide to do so, we would also need to devise strategies to operationalize the edits in numerous languages with a long history of treebank work.

We can additionally point to recent discussions within the Universal Dependencies project where the various ways to annotate English numerical expressions have been discussed.[1] Conversations such as

[1] https://github.com/UniversalDependencies/docs/issues/654

these are relevant for Universal Dependencies developers more widely, and for the sake of consistency such decisions should be at least considered for the other languages in the project. Our study also discusses some numeral types in the Uralic languages that are known, but not yet attested in the treebanks. Thereby, their description provides an important starting point for future work on these languages, during which these forms will inevitably be encountered.

2 Numerals in Universal Dependencies

In this paper, we discuss numerals in the Uralic languages. Probably the simplest approach would be to gather all numeral-type words on the basis of their *Universal part-of-speech numeral* (UPOS NUM) value or features making reference to numerals in different Uralic languages. Among the features at least `NumType` is one that would be presumed to be present with all numerals, although it also occurs widely with other parts of speech.[2] The possible, currently documented numeral types are cardinal numerals, ordinal numerals, multiplicatives, fractions, distributives, sets or collective numerals and ranges. These concepts provide a good base for a relatively elaborate and nuanced system, but at this phase the UD system appears slightly asymmetric.

Potential asymmetry might be dealt with by adding a binary for the split between numerals and counted nouns versus nouns with sequential deixis-like marking. In the Erzya, Moksha languages, sequential deixis is readily attested in combination with multiplicatives and sets, but due to the fact that ordinals only comprise three combinatorial instances in Erzya, it may strike us as fruitless to introduce a plus/minus binary for ordinal. The Erzya examples below illustrate this.

- nummod [-Ord] *vejke* ‘one’
- nummod [-Ord][+Approx] *kavtoška* ‘couple’
- nummod [-Ord][+Sets] *kavonst* ‘two pairs/sets’
- nummod [-Ord][+Dist] *kavtoń-kavtoń* ‘two-by-two’
- advmod [-Ord][+Mult] *kavkśt’* ‘twice, two iterations of the verb’
- advmod [-Ord][+Mult] *kavońkirda* ‘twofold, double the amount’
- advmod [-Ord][+Mult][+Approx] *kavkst’eška* ‘a couple of times’
- advmod [+Ord][+Mult] *omboćed’e* ‘for the/a second time’
- advmod [+Ord][+Mult] *ombońkirda* ‘a second time’
- amod [+Ord][+Sets] *ombonst* ‘a second set’
- amod [+Ord] *omboće* ‘second’
- det [-Ord][+Tot] *kavońeńek* ‘the both of us’
- det [-Ord][+Approx][+Tot] *kevet’eješkańest* ‘the approximately 15 of them’
- det [+Ord] *ombot’ks* ‘the second’

Above, we can observe that the approximatives and distributives including universal quantifiers are not associated with sequential deixis in Erzya. Whereas, sequence and range might readily be combined. In counting iterations of a predicate, Erzya shows a clear distinction between it and quantification of mass (‘twice’ and ‘twofold’ cannot be equated), but this distinction becomes less obvious when applied to a sequential deixis system. A glimpse at Komi-Permyak and Komi-Zyrian will remind us that multiplicatives may also be used in a distributive context (Rueter et al., 2020, 22). Multiplicatives, sets,

[2] https://universaldependencies.org/u/feat/NumType.html

distributives, etc. should not be distinguished from ordinals any more than they are from cardinals, since the term cardinal might readily be treated as a ZERO like nominative singular. The last three items within the list above are also exceptional as they would demand syntactic dependency ‘det’, which according to the guidelines is not allowed. Analogically, chosen conventions could possibly also be extended to the annotations of items such as English ‘both’ and Swedish ‘bägge’.

Conceivably, numerals might be divided into various categories according to their semantic use. The most predominant numeral types might therefore be associated with quantification, sequence, and entity naming. Quantification articulates distinctions in the mechanisms of counting. Singular entity counting is typified by the use of cardinals (such as in Finnish *yksi* ‘one’, *kaksi* ‘two’, *kolme* ‘three’, etc.), and there may be different marking patterns for the counted noun.

In many languages, there are standards by which the head noun of a *nummod* dependency takes special marking. In Komi-Zyrian, Komi-Permyak and Hungarian, for example, the counted noun shows no deviance from its regular nominative singular marking strategies when qualified by any cardinal numeral. In Balto-Finnic, Finnish, Estonian, Livvi and Karelian, the partitive singular marks the counted nouns when they are qualified by numerals two and above, even though their syntactic position would otherwise call for a nominative singular—for other cases a fitting semantic or syntactic case is used, i.e. phrase agrees in case.

(1) a. *kolme* *šukupolvie*
three.NOM.SG generation.PAR.SG

‘two generations’ (krl: vepkar-1652.40)

b. *kuutta* *kertua* *enemmän*
six.PAR.SG time.PAR.SG more

‘six times more’ (krl: vepkar-1740.21)

c. *šuašša* *muašša*
hundred.INE.SG land.INE.SG

‘in a hundred lands’ (krl: vepkar-1740.6)

Contrastively, the Mordvin languages, Erzya and Moksha, exhibit a variation that has yet to be researched in depth, i.e. counted nouns do not obligatorily take special marking when qualified by cardinal numerals two and upward, see Markov (1961, 42) and Rueter (2013, 107), but perhaps also in dialect studies (Ryabov, 2016; Rueter, 2016; Levina, 2021; Agafonova and Ryabov, 2021). A similar phenomenon can be observed in Moksha (Rueter, forthcoming 2022). The Saami languages attest to two different strategies: Northern Saami takes genitive singular marking of its counted nouns when qualified by numerals two and above, whereas Skolt Saami makes a three-way split, a genitive singular marking the numeral range 2–6, and the partitive marking seven and upward (with the decline in language proficiency the use of the partitive has become less certain).

Sets of entities, i.e. sets with more than single members, are counted synthetically across the languages with various strategies. In Finnish, for example, pairs of scissors are counted by using plural forms of the cardinal numerals and the NP head noun alike, e.g. *yhdet sakset* ‘one pair of scissors’ (here both the numeral and the noun it qualifies are in the plural, and unlike Russian the distinction is retained for numerals five and above, too). In contrast, Erzya has its own numeral forms typically derived in *-Onst*, hence *kavonst vasońpejeĺt’* ‘two pairs of scissors’ with the counted noun in the plural. Although numerals of the sets type are typically introduced for counting pairs, they are, in fact, often used with larger sets, such as sets of six cups and saucers.

Iterations of predications are often counted with adverb derivations of cardinal numerals, but the productivity of these derivations still requires assessment from language to language. While Finnish only minimally utilizes the word forms in *-sti*: *kahdesti* ‘twice’, *kolmesti* ‘thrice’ and *tuhannesti* ‘a thousand times’, the Hungarian, Komi-Zyrian, Komi-Permyak, Erzya and Moksha languages use regular derivations for indicating ‘X times’, *-szer/-ször/-szor*, *-ïś*, *-iś*, *-kśt’* and *-kśt’*, respectively. Needless to say, matters become confusing when these iterative numerals are categorized as multiplicatives in UD. The result, at least in Erzya, is that 'being paid *kavkśt́' = twice*' and 'being paid *kavońkirda = double* or *twofold*' are registered as the same thing, which is by no means always the state of affairs semantically, but from a syntactic perspective it is plausible.

Distributive numerals are not a simple class. They can be further categorized into subclasses, as immediately becomes apparent in the two Hungarian strategies: *két-két* 'two each' with a noun head, and *kettesével* 'two at a time' with a verb head. Whereas the former may be used as a definite numeral in the context *Berta és Rudi két-két csomagot hozott* 'Berta and Rudi brought two suitcases each', implying that a total of four suitcases were brought, the latter expression is indefinite. The indefinite distributive numeral *kettesével* 'two at a time' in nearly the same context *Berta és Rudi kettesével hozta a csomagokat* 'Berta and Rudi brought the suitcases two at a time'[3] would indicate that each iteration of the predication involves two suitcases, but there is no indication regarding the number of iterations – it could be any number of times. In this context, definiteness is lent by the object, i.e. 'the suitcases'.

Approximative numerals are numerals with values slightly less or more than the number given. Finnish, for example, attests *parikymmentä* 'about twenty' from the words *pari* 'couple' and *kymmentä* 'ten (partitive)'. In addition to constructions with the element *pari*, there are fairly regular derivations formed from other basic numerals as well: *kolmisen* + *kymmentä* 'approximately thirty'.

In Erzya, as in Moksha, approximative forms in *-ška* are found for counting entities *vet'eška lomań* 'about five people' and iterations *kolmoškakst'* 'about three times'. With the use of an approximative numeral, the likelihood rises that no plural marking is indicated on the counted noun. The predominance of nominative singular marking of the NP head also holds when the approximative is marked with an N–(N + 1) strategy, i.e. *vet'e-koto lomań* 'five-or-six people'. The use of adjacent numerals to indicate approximate values is also found in Komi-Zyrian, i.e. *vit-kvajt* and *vit-ö-kvajt* both translate to five or six.

In Finnish, the expression of range with numerals follows the same pattern as is observed in point of departure to end destination, i.e. the elative case marks the starting point, and the illative marks the end point. In the range 5–7 kilometers, the Finnish involves *viidestä seitsemään kilometriä* five+elative, seven+illative and kilometer+partitive, which is the same counted noun strategy observed in basic numerals.

Fractions in Finnish can be expressed in at least two different ways. One way is to join the ordinal nominative singular with the noun *osa* 'part', hence *viides* + *osa* = *viidesosa*, where only the end is declined and as such is distinguished from 'the fifth part' of something, where we would actually be talking of sequences. Syntactically, *neljä viidesosaa* 'four fifths' functions in the same manner as any noun with a cardinal qualifier, i.e. the NP head is marked with the partitive singular when in an otherwise nominative-singular position nummod(*viidesosaa*, *neljä*). The second derivational expression for 'fifth' is *viidennes*, it too is treated syntactically as a counted noun, as appears to be the case in other Uralic languages.

Universal quantifiers, such as the Finnish *molemmat* 'both', have more complex counterparts in Hungarian *mindkettő* (literally 'all' + 'two'), which may also take associative marking for first, second and third persons plural in *mindkettünk*, *mindkettetek*, *mindkettük*, respectively. The Hungarian *mindhárom* 'all three', 'tous les trois' then comes as no surprise, and one begins to expect subsequent *mindnégy* 'all four'. Komi-Zyrian and Erzya attest to yet another aspect: the associative personal reference can also be in the singular, allowing for access. If we are speaking of a singular 'person' and mention that 'the (lit.) three of him/her are moving to town' (Rueter, 2013), we access a definite universal quantifier pronoun with reference to this single person. This feature is not observed in Hill Mari or Udmurt (Kel'makov and Hännikäinen, 2008, 111–112). Ordinal numerals can be associated with multiplicative, iterative and sets features. This has been observed in the presentation of some morphology for Erzya, above.

Numerals appear in entity naming, for example the Finnish *viitonen* ∼ *vitonen* 'fiver' may be used when making reference to money, on the one hand, but it could also be used in reference to a street car, where we would be more likely to translate it as 'street car five' or 'street car number five'. Thus is fits

[3] cf. http://en.utdb.nullpoint.info/type/hungarian/distributive-numerals/dupldnn-sufdnv

directly into a list of problems in apposition, such as ‘the color purple’, ‘the word terrorist’ and many others including numerals discussed by Schneider and Zeldes (2021). An extension to this numeral issue is found in Finnish *viitonen* in reference to ‘house number five’, but the same `5` is transformed to the cardinal-form *viisi* if the house is `5a` or `5b` – *viisi a* or *viisi b*, respectively (no partitive, of course, so we are not counting letters). Here, the Erzya solution is to use the ordinal *vet’eće* ‘the fifth’ for `5` and *vet’eće a* ‘fifth a’ for `5a`, which results in ambiguous homonymy.

There are differences observed across languages, where synthetic versus analytic expressions of the same numerical values might be dealt with differently. Thus, our first overview discusses the largest spread of numeral types, forms across languages. Once the collection is complete, the numeral words can be classified according to the dependencies and features. In Finnish, for example, we predict four different and regular dependencies: nummod (for cardinals and plural cardinals with plurale tantum), advmod (for counting iterations of a predication, e.g. once, twice, thrice), advcl (for distributive quantification), amod (for ordinals). Other languages, it will be noted, may have extensive det (this is not really productive in Finnish, but would be the equivalent for ‘both’ and its analogues with universal quantification of numbers three and up, probably with person marking as well, e.g. ‘the two of us’).

2.1 Numeral type

According to the Universal Dependencies documentation, some numerals can be classified as adjectives and some as adverbs.[4] Thereby, in the UD guidelines both *ADV* and *ADJ* are often found as the part of speech categories for numeral expressions. At the same time, there are also situations where the NumType feature occurs with different parts of speech.

In several treebanks in the Romance languages, for example, there are pronouns such as Spanish *mucho* and *poco* which have a feature value NumType=Card. Such marking on pronouns is not common in the treebanks, although we do find English *first*, *second*, *third* and *latter* receiving POS tag *PRON* and feature NumType=Ord. This is also the style in Finnish, with *toinen* ‘second; another’ being marked similarly, and Erzya and Komi-Zyrian treebanks offer similar examples. As the combination PRON and NumType can be found only in treebanks for 10 different languages, we believe it is highly likely that similar annotations could be extended to many other languages within the project.

Nouns that are marked with NumType appear in a bit larger array of languages, all in all within 13 languages, among them, Uralic languages North Saami, Erzya and Estonian. In North Saami, these instances are collective nouns with NumType=Coll. In Erzya word *pel’* ‘half’ is marked with NumType=Frac. In Estonian the only occurrences are with gene names containing numbers, such as IL-5, where NumType=Card is attested. These are all reasonable uses of NumType, as these noun types do have countable properties that are relatively well captured by the NumType feature. But again as the solutions seem language specific the annotations could be somehow harmonized or extended to more languages.

In Finnish, Icelandic and Korean treebanks we find examples of punctuation being marked with NumType=Card. No matter how the annotation is motivated, being this rare and narrowly distributed is possibly problematic for the comparability of the languages. The Estonian treebanks EDT and EWT only use NumType with two values, Card and Ord. This does not appear to rule out fractions, but they are dealt with differently, i.e. 3/4 is given the features NumForm=Digit and NumType=Card. Of course, here the value *Digit* indicates not written as words. A second issue in EWT is that the feature `NumType=Ord` is used with both UPOS *NUM* and *ADJ*. It seems that ordinal digital numerals consisting of an Arabic numeral followed by a full stop are treated as *ADJ*, whereas automobiles from different years have an abbreviated year digit pair followed by an apostrophe. This latter type has the UPOS value *NUM*, should this be the case? We will not widely compare the differences between multiple treebanks on the same language, although we do acknowledge this is an issue that needs further attention.

Having discussed the general use of NumType feature and some rarer patterns that can be found, we will next describe more in detail different numeral types and their occurrences, with references both to Uralic and other language families, as necessary.

[4] `https://universaldependencies.org/u/pos/NUM.html`

2.1.1 Cardinal numerals

The cardinal numeral type in UD is typified as an expression for counting singular items. Thus, this feature might be associated with the UD part of speech NUM (as in one, two, three, etc.). This feature value is also used with non-numerals (as in *many*, *few*, Czech *kolik* 'how many', etc.). Here, however, individual languages make a split between use of UPOS *DET* and *NUM*. The latter of which, apparently, is defended in Czech by a strong grammatical tradition, might be used for the interrogative *kolik* 'how many', which evokes cardinal numerals. Czech includes yet a third type of words as cardinals which seem to indicate the total number, e.g. *čtvero* (as in *Čtvero ročních dob* 'The Four Season', all four), *desatero* (as in 'the Ten Commandments', all ten). This presumably explains the definition of *oba* 'both', which in Czech is marked as UPOS *NUM*, whereas Talbanken deals with *bägge* 'both' as a *DET*. And then there is the one instance of *desatero* in the treebank *Desatero investora* 'Lit. The ten investors', where the word *desatero* has the UPOS *NOUN*.

This third group of cardinals, which is not observed in Swedish as a consistent counting system, appears with a nummod dependency in Czech to match the UPOS *NUM*. In Swedish and other languages without this counting system, words with the meaning 'both' are generally dealt with as *DET*, and they have a feature *PronType=Tot*.

2.1.2 Ordinal numerals

Ordinals can be seen to represent subtypes of adjectives and adverbs. In addition to the amod dependency associated with the words *first*, *second*, *third*, there are analogical interrogatives, etc.), there is also an advmod dependency, associated with ordinal multiplicatives, such as the Czech *poprvé* 'for the first time'. By applying the feature value NumType=Ord to both UPOS *ADJ* and *ADV*, we could remove the NumType=OrdMult feature value used in Komi-Zyrian *ńol'öd* 'fourth' UPOS *ADJ* and *ńol'ödyś* 'for the fourth time' UPOS *ADV* and similarly in Erzya, Moksha and Komi-Permyak. The downside is that the parallel between cardinal and ordinal multiplicatives becomes less obvious. If we were to do so, we would be faced with the challenge of addressing numerals with three features: ordinal multiplicative and distributive.

Numerals can be classified according to what they actually enumerate or do they at all. In Erzya, the numeral type (a) *vejke, kavto, kolmo, ńil'e* is used for counting individual entities. The pertinent dependency is nummod. (b) *vejenst, kavonst, kolmonst, ńil'enst* is used for counting set entities from pairs of scissors to sets of cups. The pertinent dependency is nummod. NumType=Sets (c) *vest', kavkst', kolmokst', ńil'ekst'* is used for counting iterations of a given predication. Thus this has a advmod dependency. NumType=Mult (d) Delimiting associative collectives *śkamost, kavońest, kolmońest, ńilenest* provide universal quantification values found in the expressions 'alone', 'both', 'all three' with the addition of associative reference to number and person. These numerals are used in secondary predication with reference to the subject or object. Features include PronType=Tot (e) Distributive, imperfect *kavtoń-kavtoń, kolmoń-kolmoń, ńil'eń-ńil'eń* NumType=Dist Aspect=Imp (f) Distributive, perfect *kavtoń-kavto, kolmoń-kolmo, ńil'eń-ńil'e* NumType=Dist Aspect=Perf (g) *vejeńkirda, kavońkirda, kolmońkirda, ńil'eńkirda* has an advmod or amod dependency, and the feature value NumType=Mult.

2.2 Numeral dependencies

Among the dependency relations assigned to the numerals, the most common is *nummod*. In many Slavic treebanks an additional relation of *det* is used, as in *det:nummod*. This is not used in other treebanks. In Beja treebank there is an individual occurrence of *nummod:det*. Another subtype of *nummod*, *nummod:entity*, appears to be used only in the Russian treebanks, especially in relation to the symbol '№'. Additionally *nummod:flat* appears only in one Polish treebank. Phenomena attested and seen necessary to annotate in the Slavic languages could also be very relevant for work with the Uralic languages, many of which have been in extensive contact with Russian.

Our analysis also indicates that the relation *nummod* in the Uralic languages virtually always connects to part of speech *NUM*. With the other languages, there is extensive variation, even though this relation is always the most common. Whether this is simply a matter of annotation conventions, linguistic description traditions or actual linguistically relevant differences, remains to be studied.

3 Discussion

As we have shown, numerals and related expressions are an area for fruitful and needed further discussion in the Universal Dependencies project. Which forms all get numeric features extends widely beyond just numerals themselves, and many lexical items that have counting properties could be annotated with NumType features, and already be annotated in different treebanks. Which of the individual solutions in different treebanks should be described better in the documentation and adapted further, and which should be harmonized in comparable uses of the treebanks, remains to be discussed, but we hope our observations help at least a bit along this path. Of course, how work on various inconsistencies should or could be coordinated across the hundreds of treebanks already in the Universal Dependencies project is not entirely clear, and remains certainly a large challenge. At the same time, new treebanks are still continuously emerging, and paying attention to various strategies used in existing treebanks should help the maintainers of these new languages to adapt their conventions. When diverse language families are included, new questions inevitably arise. For example, in Apurinã there are very few actual cardinal numbers and quantification is expressed in verbal constructions (Facundes et al., 2021; Rueter et al., 2021a).

The issue how to handle Komi-Zyrian numerals was also recently discussed in the relation to Komi morphological analyser (Rueter et al., 2021b, 67), which points to the fact that the best possible annotation scheme is often a very relevant question for uses beyond the Universal Dependencies project itself. We also believe that the classification and annotation of numerals is important from the point of view of basic linguistic research and language description. As the description of Erzya counting expressions in this study showed, the system is already very complicated and nuanced in this one language, and is just starting to be adequately described in the newest grammatical descriptions (Suihkonen and Solovyev, 2013). We presume the description of many smaller Uralic languages remains much less complete, not to even mention less studied language families of the world, which also have started to have significant presence in the Universal Dependencies project. This kind of easily accessible information about counting expression at large could be immediately beneficial, for example, in typological research, and systematic annotations and documentation in projects such as Universal Dependencies is one modern way to distribute this description.

References

Nina A. Agafonova and Ivan N. Ryabov. 2021. Ulânovskoj oblasten' novomalyklinskoj raionon' Èrzân' velen' kortavkstnèsè azorksčin' nevticâ suffikstnèn' baška ionksost. In Mika Hämäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*. University of Helsinki.

Sidney Da Silva Facundes, Maríia Fernanda Pereira de Freitas, and Bruna Fernanda Soares de Lima-Padovani. 2021. Number expression in apurinã (arawák). In Mika Hämäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*. University of Helsinki.

Valentin Kel'makov and Sara Hännikäinen. 2008. *Udmurtin kielioppia ja harjoituksia*. Apuneuvoja suomalais-ugrilaisten kielten opintoja varten — Hilfsmittel für das Studium der finnisch-ugrischen Sprachen, XIV. Finno-Ugrian Society, Helsinki, Finland.

Mariâ Z. Levina. 2021. Èlektronnyj âzykovoj korpus kak faktor soxraneniâ mordovskix (mokšansogo i èrzanskogo) âzykov. In Mika Hämäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*. University of Helsinki.

F. P. Markov. 1961. Prialatyrskij dialekt èrzâ-mordovskogo âzyka (the prialtyrsk dialect of the erzya-mordvin language). In *Očerki mordovskix dialektov, tom V*, pages 7–99, Saransk, Mordovia ASSR, USSR. Mordovskoe knižnoe izdatel'stvo.

Niko Partanen and Jack Rueter. 2019. Survey of Uralic Universal Dependencies development. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 78–86, Paris, France, August. Association for Computational Linguistics.

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

Jack Michael Rueter and Francis M Tyers. 2018. Towards an open-source universal-dependency treebank for Erzya. In *International Workshop for Computational Linguistics of Uralic Languages*.

Jack Rueter, Niko Partanen, and Larisa Ponomareva. 2020. On the questions in developing computational infrastructure for Komi-Permyak. In *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*, pages 15–25.

Jack Rueter, Marília Fernanda Pereira de Freitas, Sidney Da Silva Facundes, Mika Hämäläinen, and Niko Partanen. 2021a. Apurinã Universal Dependencies treebank. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 28–33, Online, June. Association for Computational Linguistics.

Jack Rueter, Niko Partanen, Mika Hämäläinen, and Trond Trosterud. 2021b. Overview of open-source morphology development for the Komi-Zyrian language: Past and future. In *Proceedings of the Seventh International Workshop on Computational Linguistics of Uralic Languages*.

Jack Rueter. 2013. Quantification in Erzya. In Pirkko Suihkonen and Valery Solovyev, editors, *Typology of Quantification:*, LINCOM Studies in Language Typology, pages 99–122, Germany, 12. Lincom GmbH.

Jack Rueter. 2016. Towards a systematic characterization of dialect variation in the erzya-speaking world: Isoglosses and their reflexes attested in and around the dubyonki raion. In Ksenia Shagal and Heini Arjava, editors, *Mordvin Languages in the Field*, volume 10 of *Uralica Helsingiensia*, page 109–148.

Jack M. Rueter. forthcoming 2022. Mordvin. In Daniel M. Abondolo; Riitta Valijärvi, editor, *The Uralic Languages*. Routledge.

Ivan Ryabov. 2016. Ob issledovanii èrzânskix dialektov metodami lingvističeskoj geografii [on research of the erzya dialects with linguistic-geographic methods]. In Ksenia Shagal and Heini Arjava, editors, *Mordvin Languages in the Field*, volume 10 of *Uralica Helsingiensia*, page 91–108.

Nathan Schneider and Amir Zeldes. 2021. Mischievous Nominal Constructions in Universal Dependencies. *arXiv preprint arXiv:2108.12928.*

Pirkko Suihkonen and Valery Solovyev, editors. 2013. *Typology of Quantification: On Quantifiers and Quantification in Finnish and Languages Spoken in the Central Volga-Kama Region*, volume 28 of *Studies in Language Typology*. LINCOM, Munich. Quantification in Erzya, Finnish, Russian, Tatar, Udmurt, with appendices in Chuvash, English, Erzya, Finnish, Russian, Tatar and Udmurt.

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

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, H͡órunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Deniz Baran Aslan, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaur Barkarson, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Mihaela Cristescu, Philemon. Daniel, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae

Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Kaoru Ito, Tomáš Jelínek, Apoorva Jha, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Lng Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Dmitry Sichinava, Janine Siewert, Einar Freyr Sigursson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Rachele Sprugnoli, Steinh͡ór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2021. Universal Dependencies 2.8.1. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman. 2015. Slavic languages in Universal Dependencies. *Natural Language Processing, Corpus Linguistics, E-learning*, pages 151–163.

Mischievous Nominal Constructions in Universal Dependencies

Nathan Schneider **Amir Zeldes**
Georgetown University
{nathan.schneider, amir.zeldes}@georgetown.edu

Abstract

While the highly multilingual Universal Dependencies (UD) project provides extensive guidelines for clausal structure as well as structure within *canonical* nominal phrases, a standard treatment is lacking for many "mischievous" nominal phenomena that break the mold. As a result, numerous inconsistencies within and across corpora can be found, even in languages with extensive UD treebanking work, such as English. This paper surveys the kinds of mischievous nominal expressions attested in English UD corpora and proposes solutions primarily with English in mind, but which may offer paths to solutions for a variety of UD languages.

1 Introduction

Universal Dependencies (UD; Nivre et al., 2016, 2020; de Marneffe et al., 2021) is a framework describing morphology and dependency syntax cross-linguistically. It establishes common labels and structural constraints for annotating data, comparing languages, and training and evaluating parsers.

This paper, intended for readers familiar with UD (specifically, Basic Dependencies in version 2), addresses what we see as a significant shortcoming of the current guidelines: "mischievous" nominal structure—roughly, constructions that form noun phrases beyond the canonical components of determiner or possessive, adjective modifier, noun compound modifier, head noun or pronoun, modifier PP, and modifier clause. Many of these are productive but narrow constructions forming multiword names, dates, measurements, and compound-like structures.

Such expressions often buck ordinary restrictions on NP structure: Kahane et al. (2017), for instance, note that "most languages have particular constructions for named entities such as dates or titles.... These subsystems are in some sense 'regular irregularities', that is, productive unusual constructions." In other words, names and dates often do not fit the mold of other noun phrases, though as we will show below, the issues they raise pop up in other environments too. For many of these mischievous constructions, the existing UD syntactic relations are inadequate, or inadequately described, and corpora are widely inconsistent as a result—in some cases within a single treebank or between treebanks in the same language.

Many of the issues presented below have been discussed at length within the UD community but without any definitive resolution. Our goal is to consolidate the discussion and argue for a coherent approach (or set of alternatives) based on careful analyses of English constructions across a range of text types.[1] To minimize added complexity to the UD scheme, our proposals are conservative, focused on clarifying boundaries between existing labels and in some cases proposing new subtypes (which, though language-specific, may be adapted to other languages). While we refrain from proposing new universal relations that would force extensive editing across languages to maintain validity, we welcome feedback on related phenomena in other languages. Although our analysis is focused on English, we believe that similar reasoning applies to a range of other languages which cannot be adequately examined here due

[1] Some short examples in this paper come from introspection, while longer examples and statistics are taken from the English Web Treebank (UD_English-EWT; Silveira et al., 2014), and UD_English-GUM (Zeldes, 2017) or UD_English-GUMReddit (Behzad and Zeldes, 2020), which together cover a broad spectrum of spoken and written genres and writing styles.

to space reasons; we hope that guideline discussions in those languages will benefit from the analyses below.[2]

2 Name Descriptors

We turn first to proper names, especially names of persons, and the constructions by which a speaker can elaborate on a nominal referring expression.

(1) a. I met Gaspard Ulliel.
b. I met Gaspard Ulliel, the French actor.
c. I met the French actor, Gaspard Ulliel.

(2) a. I met French actor Mr. Gaspard Ulliel.
b. *I met French actor.
c. *I met the Mr. Gaspard Ulliel.

How are these handled in UD? The **`flat`** relation comes into play for open-class expressions with no clear syntactic head, canonically including personal names like *Gaspard Ulliel*. A flat structure, by convention, is represented in UD by designating the first word as the head of each of the subsequent words, which attach to it as **`flat`** (a "bouquet" or "fountain" analysis).

The trouble is that referring expressions may contain descriptors beyond personals. Following the Cambridge Grammar of the English Language (CGEL; Huddleston and Pullum, 2002), we distinguish two types of pre-name descriptors in English: An **appellation** is a title that would be used to formally address somebody by social status (e.g. occupation or gender), such as ***Mr.*** *Obama* or ***President*** *Obama*. An **embellishment**[3] is a bare nominal phrase preceding the name (and appellation if there is one) describing the referent with category information like *actor*, *French actor*, or *surprise winner of the Kentucky Derby*.[4] The embellished name may have an inanimate referent, as in ***German car maker*** *BMW*.[5] In English, embellishments are characteristic of select genres such as news.[6] (2a) contains an embellishment and an appellation within the same referring expression. The current UD guidelines state:[7]

> If the two nominals participate in denoting one entity, the default relation to connect them is **`flat`** (which may also be used to connect other nodes that are not nominals). Typical examples are personal names: we can say that *John Smith* is a special type of *John* as well as a special type of *Smith*, but none of the names governs the other and either of them can be omitted. In many languages this analysis extends to titles and occupations, as in English *president Barack Obama*.

Yet the flat analysis for embellishments and appellations yields counterintuitive results. That they are bare NPs and are omissible—whereas the personal name is not, as shown by (2b)—is strong syntactic evidence that they are modifiers. Moreover, it should be intuitively obvious that *Gaspard* and *Ulliel* form a coherent unit of structure—yet under the bouquet analysis for flat structures (i.e. attaching all children to the first token), *Ulliel* would have distinct heads for *Gaspard Ulliel*, *French actor Gaspard Ulliel*, and *Mr. Gaspard Ulliel*.[8]

Further discussion in the guidelines acknowledges treating titles as **`flat`** is controversial, but explains that titles do not meet typical criteria for **`nmod`**, **`compound`**, or **`appos`**. An **`nmod`** typically receives its own independent case marking (possessive or prepositional in English). **`appos`** is limited in UD to relations

[2] An extended version of this paper (Schneider and Zeldes, 2021) contains additional recommendations regarding numbers and adverbial NPs, omitted here due to space limitations.

[3] Also called "false title", described here as a kind of apposition: `https://en.wikipedia.org/wiki/False_title`

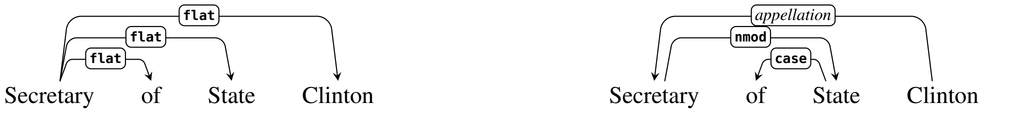
[4] An anonymous reviewer has commented on the difficulty of applying the bare nominal diagnostic in languages with different determiner systems, such as Slavic languages, Chinese, or Japanese. We fully acknowledge that equivalent constructions may look quite different in those languages, but also believe that the problems analyzed here are both substantial enough in English to merit a more detailed treatment, and common enough in other languages that the discussion is likely relevant beyond English.

[5] Thanks to an anonymous reviewer for this example.

[6] A newscaster might say, ***Surprise winner of the Kentucky Derby*** *American Pharaoh received a hero's welcome upon returning home today. . . .* Note the lack of an article at the beginning of the sentence.

[7] `https://universaldependencies.org/workgroups/newdoc/two_nominals.html`

[8] Note that some embellishments and appellations contain clear internal structure (e.g., ***French actor*** *Ulliel*—**`amod`**; ***Secretary of State*** *Clinton*—**`nmod`**, **`case`**). This does not pose an additional problem for the **`flat`** analysis, however: even dependents within a flat structure may host internal modifiers, as was recently clarified in the guidelines.



(a) Current flat analysis

(b) Proposed modifier structure for appellation

Figure 1: An appellation with current vs. proposed structures (several options for the relation label of the "*appellation*" dependency discussed below).

between two full NPs (or DPs, i.e. NPs including a determiner), as in (1b, 1c). And crosslinguistically, "titles do not usually behave like compounds: in German, they are not joined to the following words, as compounds are normally joined in German, and they appear at the beginning of names in both German and Hebrew, even though German compounds are head last and Hebrew compounds are head first."[9]

Nevertheless, we suggest that appellations and embellishments be removed from the flat analysis. Exactly how this could be achieved is considered below.

2.1 A Relation for Titles?

A narrow solution would be to group appellations and embellishments under the category of **titles**. As these constructions are frequent and distinctive, a subtype called **`:title`** might be appropriate, and subtyping could alleviate the concern that none of the existing top-level deprels is a perfect fit. Alternatively, a new top-level relation could be introduced. We thus begin by considering the following options:

- **`title`**, a new top-level relation
- **`compound:title`**
- **`appos:title`**
- **`nmod:title`**
- **`nmod:desc`**, a broader subtype, meant to cover additional mischievous nominals

A new top-level relation? A new top-level (universal) relation, **`title`**, presupposes that honorific titles, at least, occur widely across languages and may have idiosyncratic syntax. However, it seems possible that in some languages titles might have 'normal' syntax, and would not need such a top-level relation at all. Even for languages with conspicuous title syntax, UD relations aim to be as compact as possible; adding major labels is not done lightly, and would require waiting for UDv3, not to mention imposing costs on many treebank maintainers and requiring updates to existing tools. We therefore prefer subtyping an existing relation.

Problems with `compound:title`. In English, **`compound`** dependent nouns too are bare (lack a determiner of their own), similar to appellations and embellishments, suggesting a subtype **`compound:title`**. In fact, there is prior art in UD: Finnish UD documents the label **`compound:nn`** for appellations.[10]

However, there are important differences that suggest compound nominals (at least in English) and titles are two different beasts. While the definition of **`compound`** is quite vague, its applicability to modifiers of nouns is clearest in determinative compounds, either where both the head and modifier are part of a multiword proper name like *Washington Post*; or where the head denotes a kind (usually, a noun that could be made either definite or indefinite) which is restricted by the modifier, e.g. *cake flavors*. Often such non-name combinations could be paraphrased with a possessive or prepositional construction if used literally (*flavors of cake*); and often compounds behave like complex words and may become lexicalized as idiomatic multiword expressions. By contrast, appellations and embellishments of proper name heads nonrestrictively add information about an entity and might be paraphrased with "who is" or an appositive (*French actor Gaspard Ulliel* → *Gaspard Ulliel, the French actor / Gaspard Ulliel, who is a French actor*).

Morphosyntactic evidence also weighs against the compound analysis: English compound modifiers are very rarely plural, even when denoting multiple items—whereas appellations, embellishments, and appositives agree in number with their referent:

(3) a. **Presidents** Obama and Biden [appellation]

[9] https://universaldependencies.org/u/dep/flat.html#some-further-notes-on-relations-for-names

[10] https://universaldependencies.org/docs/fi/overview/specific-syntax.html#appositions-and-appellation-modifiers

b. French **actors** Ulliel and Marceau[11] [embellishment]
c. Sam and Isaac, my **brothers** [`appos`]
d. ***eggs** carton(s)

Cartons of/for multiple eggs are *egg cartons*, stripping the plural ending from the compound modifier.[12] If appellations and embellishments were special cases of the English compound construction we would expect them to resist pluralization as well, but this is not the case (3a, 3b).

Problems with `appos:title`. Part of the practical motivation for the `appos` relation is to express a semantic notion of equivalence between referring expressions, such that an information extraction system could strip out supplementary information when matching names against entities in a knowledge base. Thus *French actor Gaspard Ulliel, my hero since childhood, won an Oscar* could be simplified to *Gaspard Ulliel won an Oscar* by removing `appos` and `appos:title` dependents. From an argument structure perspective, `appos` is characterized by not adding participants to valency frames, i.e. *Gaspard Ulliel* and *my hero since childhood* both instantiate the subject of *won*.

On the other hand, `appos` is already rather complicated (spelled out in detail below, §2.3). While embellishments are sometimes categorized as appositions, there is a lack of universal agreement that appellations and embellishments qualify as appositive modifiers; other sources (e.g., Ruppenhofer et al., 2016, p. 77) view the name rather than the embellishment as the appositive phrase.

Intermediate Proposal: a subtype of `nmod`. The rationale here is that `nmod` is the most general relation for nominals modifying other nominals. (It already has subtypes, including `nmod:poss` for possessive modifiers and `nmod:tmod` for temporal modifiers.) In English, plain `nmod` dependents have case marking or prepositions, but the subtyping can signal a morphosyntactically exceptional construction, as is already the case with prepositionless `nmod:tmod`.

If we target only titles, then `nmod:title` is the least objectionable solution narrowly tailored for embellishments and appellations, given that (a) `nmod` already has other subtypes, (b) this would avoid confusion with dominant uses of `compound` and `appos`, and (c) implementing a new universal relation across treebanks would be onerous, but treebanks are allowed flexibility to diverge and innovate with subtypes. On the other hand, there are a number of other ‘mischievous’ adnominal constructions requiring a solution, which suggests that a subtype focusing only on titles may be too narrow, motivating a more general name fitting other types of descriptive modifiers, for which we will propose a new relation (called `nmod:desc`).

2.2 Other Special Types of Nominal Modification

The above discussion is limited to appellations and embellishments that precede a name. But other, less frequent constructions bear some resemblance to these:

(4) Post-name bare nominal modifiers:
a. 11-year-old Draco, **scion of the Malfoy family**, was sorted into Slytherin.
b. Oedipus, **King of Thebes**

(5) First or second person pronoun plus noun:[13]
a. We **pilots** deserve a pay raise.
b. You **guys** deserve a pay raise.[14]

In (4, 5), the bolded nominal phrase can be omitted while its head (underlined) cannot. (4a) can be considered a post-head embellishment, and (4b) a post-head appellation. The construction seen in (5),

[11] It is unclear whether non-coordinated names referring to multiple individuals could license plural embellishments via semantic number agreement: *An argument broke out between married actors Brad and Angelina / ?married actors Brangelina / ?British comedians Monty Python*.

[12] For exceptional pluralized modifiers in Germanic compounds see also Fuhrhop (1996).

[13] Elsewhere the pronouns are analyzed as determinatives (Huddleston and Pullum, 2002, p. 374), but we deem it impractical to extend `det` to include such specialized uses of personal pronouns.

[14] The expression *you guys* has been conventionalized in some dialects as a gender-neutral second person plural.

headed by a pronoun, is a cousin of the pre-head embellishment, as shown by the third person paraphrase of (5a): ***pilots*** *Earhart and Lindbergh*. A broad relation `nmod:desc` for the special cases seen above as well as appellations and embellishments would separate them from the `appos`, `compound`, and `flat` cases while covering sufficient ground to merit its inclusion.

2.3 More on Appositives

A classic example of an appositive appears in (6). The appositive phrase, *my brother*, is a nonrestrictive full NP descriptor of *Sam*. It is syntactically omissible, and could in fact replace its head as they share the same referent. A similar phenomenon appears in (7), where an indefinite NP ascribes a property to *Sam*:

(6) Sam, **my brother**, is very tall.

(7) Sam, **a musician**, is very tall.

The current definition of the `appos` relation establishes the following criteria:

(8) An appositive (`appos`) must be
  a. a full NP
  b. modifying an NP in a reversible fashion (modulo punctuation)
  c. to the right
  d. with no intervening words.[15]

While appositive phrases are often separated by commas or parentheses, this is not a strict requirement, and of course spoken language has no commas. We understand the definition to also include:

(9) a. my brother **Sam**
  b. the color **purple**
  c. the word "**terrorist**"
  d. the play ***Much Ado About Nothing***

Cases resembling appositives in some but not all of the above respects require clarification. The bare modifiers discussed above are sometimes considered appositives, but UD excludes them with criterion (8a). (10) satisfies criteria (8a, 8c) but not (8b, 8d), whereas (11) satisfies (8a, 8b, 8d) but fails (8c):

(10) "Maybe she really does just need a little space...," Amy said, **ever the optimist**.[16]

(11) **A new Pakistani leader**, he is intent on instituting reforms.

There seem to be two ways forward:

- Relax `appos` criteria either in general or in a subtype. In particular, relaxing (8b–8d) would allow `appos` to cover (10, 11). This would contrast with `nmod:desc` suggested above, which covers bare nominal modifiers.
- Maintain the `appos` criteria in (8), and classify examples such as (10, 11) as `dislocated`. These constructions are not quite classic dislocation constructions,[17] but they could be treated as if removed from their normal apposition location.

In the interest of maintaining the status quo for appositions, we favor the latter solution and recommend using `dislocated`.

		head	modifier optional?	invertible?	agreement?	type	relation
(2a)	actor Ulliel	R	Ulliel	*Ulliel, actor	actors Ulliel and Marceau	name (head)	←**nmod:desc**
§2	President Obama	R	Obama	*Obama, President	Presidents Obama and Biden	name	←**nmod:desc**
§3.1	Church Street	R	*Street / the street	*Street, Church	Church and River Streets	name	←**compound**
(12)	Lake Michigan	L	*Lake / the lake	*Michigan, Lake	Lakes Michigan and Ontario	name	**compound**→
(14)	Figure 4	L	*Figure / the figure	*4, Figure	Figures 4 and 5	name w/ num	**nummod:name**→? **compound**→? **nmod:desc**→?
(13)	Firefox 58.0	L	Firefox	*58.0, Firefox	*Firefoxes 58.0 and 59.0	name w/ num	**nummod:name**→? **flat**? **nmod:desc**→?
§3.7	London, UK	L	London	*UK, London	*Londons, UK and Ontario	name	**nmod:npmod**→
	Joe Biden	–	(flat)	(flat)	*Joe and Jill Bidens	name	**flat**
(6)	my brother Sam	L	my brother	Sam, my brother	my brothers Sam and John	name (mod)	**appos**→

Table 1: Constructions involving names and their syntactic properties.

3 Further Issues with Names

3.1 Syntactically analyzable proper names

Several other aspects of the syntax of names need to be addressed. The syntactic properties of many of the constructions at issue are summarized in table 1. We begin by underscoring UD's policy of analyzing the internal structure of names with ordinary syntax where possible, regardless of the semantic status of the name. For example, *Church Street* is analyzed with **compound**; and *New York City* consists of an adjective which modifies a noun (**amod**), which in turn modifies another noun (**compound**).[18]

3.2 Cardinal directions

Cardinal direction modifiers of nouns (*north*, *northeast*, etc.) are annotated inconsistently in English UD corpora. Based on the tagging tradition of LDC corpora, these should be treated as nouns unless they bear overt adjectival morphology (*northern*, etc.). Cardinal direction nouns premodifying nouns should therefore attach as **compound**, whether the expression is a proper name (*North Carolina*) or not (*north coast*). When multiple parts of a cardinal direction term are separated by a space or hyphen, they are joined with **compound**: e.g. *north east* 'northeast'.

3.3 Names beginning with an entity type

Many proper names incorporate a transparent entity type. In *the Thames River*, the name is constructed as an ordinary endocentric compound, with the entity type last and serving as the head and an identifier as the modifier.[19] But *the River Thames* (along with the other examples in (12)) poses a problem as the order is reversed:

(12) a. Mount Fuji
b. Fort Knox
c. Lake Michigan
d. the River Thames

It can be argued that the head in (12d) is then *Thames*, as *River* can be omitted: *the Thames* (Huddleston and Pullum, 2002, pp. 519–20). However, this omission of the entity type could be viewed as a shortening

[15] An exception to this constraint is already found in languages with so-called Wackernagel particles, such as Classical Greek or Coptic, which appear in the second position in the sentence and can interrupt any phrase or dependency; see Zeldes and Abrams (2018).

[16] *The Body in the Casket: A Faith Fairchild Mystery*, Katherine Hall Page, 2017

[17] The preferatory appositive in (11)—which features a description followed by a definite NP, and would be perfectly at home in a newspaper—is not to be confused with hanging topic left-dislocation with a pronoun referring back to the dislocated element, as might be uttered in conversation: *My dad, he is always running late.*

[18] Previously, POS tags in the English treebanks followed Penn Treebank tags and treated all content words within a proper name as PROPN, but this was changed in v2.8; PROPN is now limited to nouns.

[19] Other place names headed by an entity type and exhibiting ordinary syntax include *Mirror Lake*, *Ford's Theatre*, and *the Dome of the Rock*.

not unlike reducing *Fenway Park* to *Fenway* on the assumption that the speaker is able to identify the referent based on the more specific part of the name. Such shortenings will vary in felicitousness depending on the particular name and context. (Plain *Michigan* does not refer to the same thing as *Lake Michigan*.)

Note also that the name-initial entity types may be pluralized when grouping together multiple entities of the same type, which distinguishes them from flat structures or typical compound modifiers and suggests they may be heads: ***Lakes** Michigan and Ontario* (cf. *Mirror and Swan **Lakes***). This fits with the expected semantics, as noun-noun compounds tend to be headed by the superordinate category, and historically it is possible that the construction is in fact a remnant of left-headed compounding from Romance place names, possibly from Norman toponym patterns (English *Mount* X, French *Mont*-X, e.g. *Mont-Saint-Michel*).

We therefore consider the examples in (12) as inverted (left-headed) compounds.[20] The identifier can attach to the entity type as **`compound`** to reflect the inverted word order in these kinds of names.

3.4 Numbered entities

Numbers can also figure into names. They can disambiguate multiple of a series of related entities named by a proper noun, as in (13). These are appendages to a proper name, syntactically omissible (with a resulting broadening of meaning), and could be treated as modifiers. Numbers can also follow an entity type, as in (14).

(13) a. Firefox (version) 58.0
b. Richard III
c. *Toy Story 3*
d. 1 Corinthians
e. World War II

(14) a. Figure 4
b. room 11b
c. pp. 5–10
d. subpart (e)
e. item (number) 3
f. *Symphony No. 5*

The cases in (14) use the number to identify a specific instance of the type. The entity type appears first, similar to the inverted **`compound`** examples in §3.3. It is a completely different construction from quantity modification, the predominant application of **`nummod`**, as in *3 items* (plural!) or *3%*. A morphosyntactic difference between the numeric modifier constructions in (13) and (14) is that only the latter exhibit agreement: *page 5* (one page), *pages 5–10* (multiple pages), but **Firefoxes 58.0 and 59.0*.

We see three options, each with pros and cons:

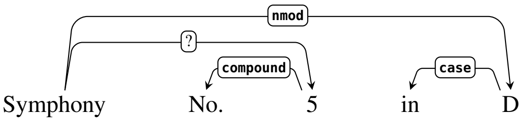
- The morphosyntactic difference notwithstanding, treat (13) and (14) as essentially the same construction, with a new relation such as **`nummod:name`** (consistent with the fact that the superordinate category **`nummod`** is currently applied to numeric modifiers generally).[21] Advantages are that (13) and (14) look very similar, and numbers are a salient property for annotators or corpus users to notice when selecting the appropriate relation. However, adding a subtype for a relatively narrow and infrequent phenomenon is questionable, and some cases are not numeric (*Level B*).
- Treat (13) and (14) as instances of more general constructions. The construction in (14) can be considered an inverted compound like *Lake Michigan* (§3.3). Flat structures could apply to the names in (13) as this construction is less morphologically transparent. This would avoid a new subtype but also may be seen as splitting hairs based on a subtle morphosyntactic criterion.
- A third option is to adopt **`nmod:desc`** for the constructions in (13) and (14). This would essentially restrict the definition of **`compound`** to substantive lexical material excluding numbering designators; **`nmod:desc`** would broadly cover miscellaneous modifiers associated with names that do not fit the more conventional constructions. This solution eclipses the similarity between *Lake Michigan* (which would remain **`compound`**) and *Figure 4*, but it perhaps avoids a counterintuitively broad application of **`compound`**. It also means that the scope of **`nmod:desc`** is a bit broader, including not just modifiers

[20] Another analysis we considered was to treat the entity type as an **`nmod:desc`** modifier, giving *Lake Michigan* the same structure as *Dr. Livingstone* or *actor Ulliel*. But the entity types in (12) seem more central to the name than titles, and are not as freely omissible, so we are not persuaded that they are modifiers.

[21] The choice of subtype parallels **`flat:name`**—an optional subtype not currently implemented in English corpora, though it is used for a number of corpora in other languages. The **`flat:name`** guidelines currently include *Formula 1* as an example; this would become **`nummod:name`** in this option.

that are secondary to the main part of a name, but also modifiers that are essential to it (just *Figure* is not a name, whereas *Ulliel* is).

(13a, 14e, 14f) illustrate a construction in which a word like *number* or *version* may precede a number to clarify that it is an identifier rather than a quantity. In modern usage this would generally remain singular even if referring to multiple items (*items number 3 and 4*), so we analyze *number* as a **compound** modifier by default, and **nmod:desc** only if plural (*items numbers 3 and 4*). "?" is provided as a stand-in for the relation between the entity type and the number given the above uncertainty:[22]



For hyphenated numeric ranges (14c), the prevailing policy in UD corpora has been to analyze the second part like a prepositional phrase *to 10*, thus an **nmod** of *5*. One of the authors takes the view that a coordination analysis would be more natural. In any event, *5* attaches to *pp.* as a modifier.

3.5 Business and personal name suffixes

Adjective-expanding suffixes like *Inc.* ("incorporated") in *Apple Inc.* should attach as **amod**. Nominal suffix designations that do not head the name, e.g. *LLC* ("limited liability corporation"), should attach as **nmod:desc**. For personal names, the suffix type *III* in (13b) is addressed above. Generational name suffixes that do not use numerals, like *Richard* ***Jr.*** and *Richard* ***the Third***, are treated as postmodifying **amod**. Other abbreviated name suffixes that would expand to nominal expressions, such as professional or honorary designations (*MD*, *O.B.E.*), attach as **nmod:desc**.

3.6 Nicknames and parenthetical descriptors

A nickname that takes the form of a full NP appended to a name, e.g. *Richard* ***the Lionheart***, can be attached as **appos**. The same goes for works of art featuring a formulaic name followed by a nickname: *Symphony No. 5 "**Fate**"*. Parenthetical descriptions following a name that are not alternate references to the entity should be treated as **parataxis**: *Pierre Vinken, **61**, said...*; *Vinken, **61 years old**, said...*; *The Chicago Manual of Style, **17th edition***; *Biden **(D)** said...* (but *Biden, **a Democrat**, said...* would be **appos**).

3.7 Addresses

A street address like *221b Baker St.* is headed by *St.*, with *Baker* attaching as **compound**, and *221b* per the policy on numbered entities (§3.4). Frequently, place descriptions specify a locale-NP postmodifier without a connective word besides punctuation. Examples: *London, **UK***; *University of Wisconsin–**Madison***; *CSI: **Miami***. These should be considered adverbial NPs, which arguably should fall under the **nmod:npmod** relation.[23]

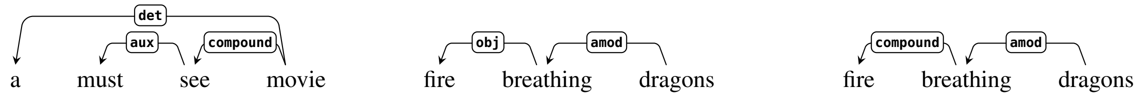
Multiple tokens of a single phone number should be joined with **flat** (this is the practice in the GUM corpus; EWT currently favors **nummod**). Separate pieces of metadata that are juxtaposed in an extralinguistic fashion (e.g., name, street address, city, postal code) should be treated as items of a list—successive items should attach to the first as **list**.

4 Phrasal Attributive Modifiers

In English, the attributive modifier position before the noun head in a noun phrase is not limited to adjectives/adjective phrases (*very easy to use*) and nominals. It also accommodates phrases like:

[22] Confirming native speaker intuitions, a search of COCA (Davies, 2010) reveals that the plural is much less frequent than the singular in the pattern N.PL *number(s)* NUM *and* NUM, with the exception of the abbreviated spelling, where *nos.* is more prevalent in this context than *no.* (the abbreviations seem to be especially conventional in proper names like *Symphony No. 5*).

[23] Currently, corpora sometimes use **nmod:npmod** and sometimes use **appos**, which is not appropriate as the two parts of the location are not interchangeable. Space does not permit full discussion of **nmod:npmod** here (but see Schneider and Zeldes, 2021, §6).



(a) Aux+V modifier **(b)** 'Deep structure' N+V modifier analysis **(c)** Proposed 'surface structure' analysis

Figure 2: Phrasal attributive modifiers (hyphen tokens omitted for brevity).

(15) a. a **high-quality** product
b. a **by-the-book** strategy
c. a **fly-by-night** operation
d. a **have-your-cake-and-eat-it-too** plan
e. a **come-to-Jesus**, **do-or-die** moment
f. a stern **don't-mess-with-me** look
g. a **must-see** movie
h. **fire-breathing** dragons
i. the **Bible-thumping**, **church-going** faithful
j. many **so-called** libertarians
k. a **cost-effective**, **nuclear-free** future

Assuming that the hyphenated expressions are tokenized as separate words, UD annotators are confronted with two issues: how to analyze these phrases internally, and which dependency relation to use for the modification of the external noun.

Some of the hyphenated expressions in (15) are clearly lexicalized; others are productive combinations. Expressions of this type might loosely be described as 'compounds', in the sense that the joining of multiple content words into one lexical item is the morphological process of compounding. Should the hyphenated parts thus be joined together with **compound** across the board? We are hesitant to establish this policy because it would overload an already very broad relation label. Centrally, in noun phrases, **compound** describes modification of a noun by another noun. If it applies to the examples in (15), it would be for attachment to the underlined noun, not the internal structure of the hyphenated expression.

Another consideration is that the internal structure of the hyphenated phrases is *largely* regular: phrasal modifiers of nouns can be structured as modified nouns (15a), PPs (15b), VPs (15c, 15d), imperative sentences (15e, 15f), and verb clusters (15g). These structures are transparent, and just as UD policy analyzes regular internal structures in proper names like *University of Wisconsin*, we advocate recognizing internal structure here.

Yet synthetic or argument structure compounds such as *fire-breathing*, *Bible-thumping*, and *church-going* (15h, 15i) invert the normal clausal order. Neither *fire* nor *Bible* nor *church* is the subject in the clausal paraphrase: *fire* is the direct object in *breathing fire*; the paraphrase of *Bible-thumping* would require reordering and adding a determiner or plural for the direct object; and the paraphrase of *church-going* would require a preposition: *going **to** church*. Meanwhile, *so-called* (15j) lacks any obvious paraphrase as a clause. We take these anomalies in word order and morphosyntax as clear evidence that left-headed 'deep structure' VP material is being grafted onto a right-headed compound in the 'surface structure'. As Basic UD aims to represent surface syntax, we join these expressions as **compound**, as shown for *fire-breathing* in figure 2c (vs. figure 2b). The adjective-headed combinations in (15k) should also use internal **compound**, as should numeric modifier compounds like *a **10-year** plan*.[24]

The next question is the external attachment, which is made difficult by UD's lexicalist principle that the part of speech of a word determines which relations it can participate in. Consider *must-see* (15g), which is not a full VP, merely an auxiliary plus its head verb. Is this to be treated as a clausal dependent—**acl**, or even **acl:relcl** (a relative clause)? This seems dubious; note that a relative clause paraphrase would involve an embedded subject, e.g. *a movie that **one** must see*, or else a passive—*a movie that must be seen*. It is also doubtful whether (15c–15f) should be treated as clausal modification, yielding several different dependency labels for the attributive relationship. A simpler solution, it seems to us, is to treat attributive phrasal expressions internally headed by verbs like coerced noun phrases,[25] with **compound** for the external attachment, as shown in figure 2a. As for PP modifiers like in (15b), it seems simplest to attach

[24] Contrast *10-year* (**compound**) with *10 years* (**nummod**), where the number modifier controls agreement.

[25] Kahane et al. (2017) suggest expanding the UD notion of multiword token to include idiomatic phrasal expressions, separating their external syntactic behavior from their internal structure. This would make it convenient to represent the expression *must-see* as a multiword NOUN comprised internally of an AUX and a VERB. This could be indicated via a morphological feature ExtPos=NOUN on the internal head, *see*.

them as **compound** rather than **nmod**; on this view, English nominal **compound** is equivalent to attributive modification by a non-possessive nominal phrase (a hypothetical alternate name being **nmod:attr**).

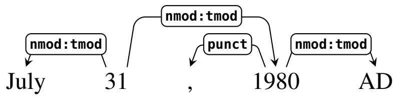
To summarize, our proposed policy for phrasal attributive modifiers of nouns is:

- The attributive expression is internally analyzed with regular relations to the extent possible, except where those relations defy ordinary word order or morphosyntax. **compound** is used internally for anomalous relations.
- In the interest of simplicity, all non-possessive attributive modifiers attach as either **compound** if internally headed by a nominal or nominalized phrase (including PPs), and **amod** etc. for adjectival heads, as appropriate.

5 Dates

While analytically expressed dates like *the thirty-first of July* follow normal syntax (with *thirty-first* elliptical for *thirty-first day*), there are special written formats for dates and times. Instead of a flat structure, which would obscure the compositionality of dates, we propose the simple principles of (a) treating the most precise part of the expression as its head, and (b) connecting the parts of the expression together with **nmod:tmod**.[26]

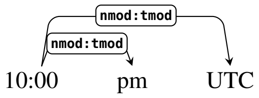
For example, *July 31, 1980 AD* consists of a year expression (*1980 AD*) and a month both modifying a date:



Another convention puts the date before the month (*31 July*). There, too, the date would be the head. Even when the date is written as an ordinal—*July the fourth*—the month should be considered a temporal modifier because it can be omitted with sufficient context (*I'll see you on the fourth*; **I'll see you on July*). This is in contrast to *Richard the Third* (§3.5), where *Richard* is the head.

A further practical consideration is that UD tree heads are often used to determine minimal token spans for annotations such as entity recognition, mentions in coreference resolution, and entity linking spans for Wikification (associating mentioned entities with their Wikipedia entries; Ratinov et al., 2011). Such minimal or 'MIN' spans (Poesio et al., 2018, p. 12) are then used for training and scoring systems in 'fuzzy' match scenarios. It makes intuitive sense for the day in date expressions to form the minimal span which needs to be identified, since the other tokens, i.e. years and months, already form the minimal spans for the nested mentions of those years and months as separate entities. This use of UD-tree heads is already in place for non-UD corpora using UD parses, such as ARRAU (Uryupina et al., 2020), and in the gold standard UD English GUM for NER, coreference and Wikification (Lin and Zeldes, 2021).

For time expressions we follow similar reasoning, with an example as follows:



The time zone could alternately be expressed as a phrase like *London time*, which we would also view as **nmod:tmod**. If written as *ten o'clock*, the token *o'clock* is considered an adverb and **advmod** of *ten*. This also corresponds to an etymological reading of *o'clock* (< *of clock*), since a univerbized prepositional phrase is equivalent to an adverb (cf. adverbs like *ashore*, formed with the Old English preposition *an*, the stressed equivalent of *on*).

Zeman (2021) likewise proposes a standard for dates and times (considering English as well as Czech, Indonesian, and Chinese). That approach is similar, differing mainly in treating the year in a date expression as headed by the month rather than the date—*1980* would be a dependent of *July*, which would be a dependent of *31*, in *July 31, 1980*. While semantically intuitive (smaller units of time head the next larger containing unit), it is not clear that there is any *syntactic* motivation to group the month and

[26] We considered finer-grained relations like **nmod:month**, **nmod:year**, **nmod:era**, **nmod:ampm**, and **nmod:tz** but concluded these were too detailed for UD and should fall under the purview of information extraction.

year together. Although the month cannot normally be omitted while retaining the year, an expression like *the 31st, 1980* is only semantically nonsensical, or at best pragmatically anomalous, but not truly ungrammatical. As evidence for this we consider the possibility of felicitous day+year expressions, such as *New Year's Day 2000* (the same as 2000-01-01) or *Pentecost 2022* (2022-06-05). The year-modifies-month approach also has the disadvantage of creating nonprojectivity if the date is written between the month and the year.

Zeman (§5) suggests **appos** to link a date with a day of the week, as in *Wednesday, July 31*. We agree with this policy. Though the day of the week conventionally comes first in English, we recognize that the order may be reversed on occasion (reversibility is a definitional criterion for **appos**, which is always left-headed). Moreover, this does not affect preposition choice, as *on* marks days of the week as well as dates, supporting the **appos** analysis in which they are essentially interchangeable full NPs.

6 How prevalent are these issues?

Some readers may wonder how common the issues raised in this paper actually are, and in particular whether their frequency merits adding relation subtypes such as **nmod:desc**. Table 2 gives statistics for some types of constructions that would be covered under the umbrella of such a relation. Although the phenomena are not extremely frequent, the total token count of 373 out of 152K tokens in the UD v2.9 edition of GUM puts a putative relation covering these at rank 35 of 49 relation labels (including subtypes), between **obl:tmod** (362 tokens) and **nmod:tmod** (399), suggesting that these are not particularly rare occurrences. We also presume that depending on genre, some subtypes may become much more frequent, such as company suffixes or even personal titles—for example, the frequency of just company suffixes in EWT seems is about 2.5 per 10K tokens, compared to 0.3 per 10K tokens in GUM (other categories are harder to identify, since their annotation in EWT currently varies or is not easily distinguishable, as in the case of numbering modifiers).

construction	most frequent types	tokens (GUM)	types (GUM)
title/profession	General (15), Mr. (10), St. (8)	202	78
numbering	Figure (31), Method (20), Wave (10)	162	63
company	Inc (4)	4	1
entity type	Mount (1), Camp (1), Team (1)	5	5
total		373	147

Table 2: Frequencies of some mischievous nominal constructions in GUM.

Although adding a new labeling distinction in the form of **nmod:desc** would doubtless require some manual disambiguation effort, we feel that by surveying the constructions in this paper in detail, it becomes more feasible to design high recall, automatic approaches to creating an initial updated version of UD English with a more nuanced treatment of these mischievous constructions, using UD editing libraries such as DepEdit (Peng and Zeldes, 2018) or Udapi (Popel et al., 2017), which can then be subjected to a manual filtering pass.

7 Conclusion

Above we have reviewed many constructions involving names, values, and compounds that have pointed to blind spots in the current guidelines for the **nmod:***, **compound**, **flat**, **appos**, and **nummod** relations. We have laid out several options for improving the treatment of these constructions via clearer and more principled guidelines. The proposed improvements are of a surgical nature, minimizing disruption to other UD conventions (no new universal relations are proposed, for instance). We are cognizant that considerable effort may be required to fully revise existing UD treebanks, but note that treebanks are already inconsistent; clearer guidance can only help. Subtypes remain officially optional—it is not necessary for a treebank to distinguish subtypes of **nmod** to be compliant with the UD standard.

We invite feedback on these proposals from the UD community, particularly with regard to other languages. We are aware that treebanking efforts in other languages have encountered some of the same issues, but we have not systematically investigated our proposed solutions beyond English.

Acknowledgments

Though the synthesis of problems, argumentation, and recommendations presented above are new, many of the specific challenges and alternatives arose from background discussions on GitHub about the UD guidelines. These discussions took place over several years in more than a dozen discussion threads involving (at least): Aryaman Arora, Colin Batchelor, Flavio Massimiliano Cecchini, Xinying Chen, Çağrı Çöltekin, Sylvain Kahane, Chris Manning, Bohdan Moskalevskyi, Joakim Nivre, Martin Popel, Alexandre Rademaker, Livy Real, Jack Rueter, Sebastian Schuster, Francis Tyers, Jonathan North Washington, and Dan Zeman. We are grateful to all who provided insights.

References

Shabnam Behzad and Amir Zeldes. 2020. A cross-genre ensemble approach to robust Reddit part of speech tagging. In *Proc. of the 12th Web as Corpus Workshop*, pages 50–56, Marseille, France.

Mark Davies. 2010. The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and Linguistic Computing*, 25(4):447–464.

Nanna Fuhrhop. 1996. Fugenelemente. In Ewald Lang and Gisela Zifonun, editors, *Deutsch - typologisch*, pages 525–550. de Gruyter, Berlin.

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

Sylvain Kahane, Marine Courtin, and Kim Gerdes. 2017. Multi-word annotation in syntactic treebanks - Propositions for Universal Dependencies. In *Proc. of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 181–189, Prague, Czech Republic.

Jessica Lin and Amir Zeldes. 2021. WikiGUM: Exhaustive entity linking for Wikification in 12 genres. In *Proc. of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 170–175, Punta Cana, Dominican Republic.

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

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 *Proc. of LREC*, pages 4027–4036, Marseille, France.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: a multilingual treebank collection. In *Proc. of LREC*, pages 1659–1666, Portorož, Slovenia.

Siyao Peng and Amir Zeldes. 2018. All roads lead to UD: Converting Stanford and Penn parses to English Universal Dependencies with multilayer annotations. In *Proc. of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 167–177, Santa Fe, New Mexico, USA.

Massimo Poesio, Yulia Grishina, Varada Kolhatkar, Nafise Moosavi, Ina Roesiger, Adam Roussel, Fabian Simonjetz, Alexandra Uma, Olga Uryupina, Juntao Yu, and Heike Zinsmeister. 2018. Anaphora resolution with the ARRAU corpus. In *Proc. of the First Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2018)*, pages 11–22, New Orleans, LA.

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

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proc. of ACL-HLT*, pages 1375–1384, Portland, Oregon, USA. Association for Computational Linguistics.

Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, Collin F. Baker, and Jan Scheffczyk. 2016. FrameNet II: extended theory and practice.

Nathan Schneider and Amir Zeldes. 2021. Mischievous nominal constructions in Universal Dependencies. *arXiv:2108.12928 [cs]*.

Natalia Silveira, Timothy Dozat, Marie-Catherine De Marneffe, Samuel R. Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proc. of LREC*, pages 2897–2904, Reykjavík, Iceland.

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Francesca Delogu, Kepa J. Rodriguez, and Massimo Poesio. 2020. Annotating a broad range of anaphoric phenomena, in a variety of genres: the ARRAU Corpus. *Natural Language Engineering*, 26:95–128.

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

Amir Zeldes and Mitchell Abrams. 2018. The Coptic Universal Dependency Treebank. In *Proc. of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 192–201, Brussels, Belgium.

Daniel Zeman. 2021. Date and time in Universal Dependencies. In *Proc. of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*.

Date and Time in Universal Dependencies

Daniel Zeman
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University, Prague
zeman@ufal.mff.cuni.cz

Abstract

We attempt to shed some light on the various ways how languages specify date and time, and on the options we have when trying to annotate them uniformly across Universal Dependencies. Examples from several language families are discussed, and their annotation is proposed. Our hope is to eventually make this (or similar) proposal an integral part of the UD annotation guidelines, which would help improve consistency of the UD treebanks. The current annotations are far from consistent, as can be seen from the survey we provide in appendices to this paper.

1 Introduction

The label of the UTC time zone[1] suggests that time can be coordinated and universal. Unfortunately, date and time expressions in the world's languages are not universal, and their current annotation in the various corpora in Universal Dependencies (UD) (de Marneffe et al., 2021)[2] is far from coordinated. One likely reason is that from the point of view of a grammarian, date and time expressions are a rather marginal phenomenon. Similarly, they are not the first thing to be covered by corpus annotation guidelines; and sometimes there are no guidelines for them at all. To the best of our knowledge, this is the case of Universal Dependencies, at least of the universal part of the UD guidelines[3] (we cannot exclude that one of the language-specific sections discusses these expressions). The issue has been discussed in the UD Github issue tracker[4,5] but the discussion did not result in a concrete specification in the guidelines. No coherent proposal seems to have emerged, neither on the website nor in UD-related papers (de Marneffe et al., 2021; Nivre et al., 2016; Nivre et al., 2020). A noteworthy exception is the recent proposal by Schneider and Zeldes (2021, Section 5), who try to solve dates in English.

The main research question is whether (or to what extent) date and time expressions have internal syntactic structure. UD is a syntactic framework, so in clear cases of syntactic structure we should annotate it analogously to similar constructions elsewhere in the language. On the other hand, date and time expressions are frequent in certain genres across the world's languages, with globally understood semantics, so it would be beneficial for language-understanding applications to always organize the corresponding items (year, month, day, hour, minute…) the same way. Ideally, we would like to find a rule that is language-independent, yet it does not clash with morphosyntax when applied to concrete languages.

Date and time expressions are difficult not only because of their (understudied) grammatical peculiarity, but also because of the way they are encoded in written language. If the expression involves digits and symbols, the general guiding principle in UD is that the analysis should be parallel to how the expression is pronounced and how it would appear in a treebank of spoken language (de Marneffe et al., 2021, p. 285). Therefore, another research question of this paper is to what extent this principle can actually be followed, as in some cases there are multiple possible readings.

[1] UTC = Coordinated Universal Time

[2] We work with UD 2.8, http://hdl.handle.net/11234/1-3687.

[3] https://universaldependencies.org/guidelines.html (all webs retrieved on October 4, 2021)

[4] https://github.com/UniversalDependencies/docs/issues/113

[5] https://github.com/UniversalDependencies/docs/issues/210

In the present study, we survey various time-related expressions in a selection of the UD languages and their internal syntactic structure (if any). We then propose a solution: a set of guidelines that — if added to the UD documentation — would make annotation of these expressions easier and hopefully more consistent. The current annotation in UD treebanks is not discussed directly in the survey, but for basic date expressions, an overview is provided in the appendices. With typological diversity in mind, we conduct the research on 5 languages from 4 different families.

2 Tokenization

If a date refers to the month by its name, then the day and year numbers are separate tokens, too. However, if the date consists entirely of numbers and punctuation, some UD treebanks prefer to treat the date as a single token. This is not exactly wrong (and it helps avoid some of the issues we are going to discuss in this paper), yet we would argue that splitting the date into multiple tokens can increase the parallelism with dates where the month is named. Furthermore, the orthographic rules in some languages allow writing numerical dates with or without spaces,[6] which would lead to inconsistent annotation if the dates without spaces are not split.

Punctuation separators such as slashes or hyphens should be independent tokens, too. However, if there are periods that, according to the language-specific rules,[7] mark the numbers as ordinal numerals, we recommend to keep them in the same token as the number. The token will then be recognizable as an ordinal numeral, and it will be parallel to English *1st, 2nd, 3rd, 4th…*

It is less obvious whether a similar argument should be made for time expressions. The core part, hours and minutes, are typically spelled as one string, looking like a decimal number (although they actually use the sexagesimal system, and sometimes a different punctuation symbol). There are multiple options how to pronounce it, but it is not uncommon that the hour and minute parts are simply read as a sequence of two numbers. We suggest to keep such numbers as one token.[8] However, if the string contains unit names or abbreviations, it should be better split into multiple tokens: *19h15m* would become *19 h 15 m.*

3 Tags and Features

Names of months and days of week are considered proper names (`PROPN`) in some languages and common nouns (`NOUN`) in others. Sometimes the distinction is just in language-internal orthographical rules (whether or not the word is written capitalized), sometimes there may be deeper consequences, e.g., whether the word is used with a definite article. The decision has to be made on a per-language basis; if there are no reasons supporting `PROPN`, we suggest to use `NOUN` as the default.

Tokens consisting entirely of digits (i.e., years and sometimes days) should always be tagged as cardinal numerals (`NUM`). However, sometimes the form of the number clearly indicates that it should be pronounced as an ordinal numeral *(5^{th}, 5.)*. Ordinals are a subclass of adjectives in UD, hence the `ADJ` tag should be used. Note that the POS category is not changed to noun when the numeral heads a nominal. According to the UD tagging principles, the nominal function is encoded in the incoming dependency relation but the POS tag stays the same.

4 Dates

In the UD taxonomy of syntactic units, dates are nominals. Most often they function as temporal oblique modifiers in clauses (`obl:tmod`), as in *The data will be released on November 15, 2021*, or even without a preposition *(It took place last April)*. However, dates can appear in all other constructions where nominals can. They can be non-temporal modifiers *(What did they say about June 30?)*, subjects *(June 30 suits me perfectly)* etc.

[6] For instance, in Czech, the default is with spaces after periods but the standard admits an alterative format without spaces in business and technical documents (`https://prirucka.ujc.cas.cz/?ref=160&id=810`).

[7] E.g. in Czech, German and Finnish.

[8] It could be argued that the read-out-loud rule leads to multiple tokens for *7:15* because *seven fifteen* is written as two tokens. However, we do not think it would be useful to extend this rule to purely numerical strings because then we would have to split also numbers that do not denote time.

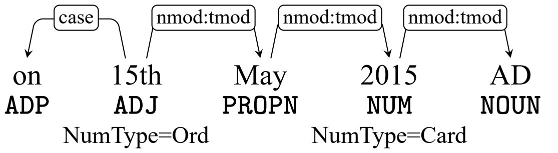
4.1 English

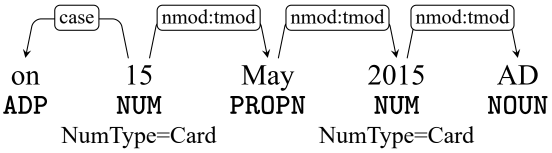
Two date patterns are common in English: *May 15, 2015* (pronounced *May (the) fifteenth, twenty fifteen / two thousand (and) fifteen*) and *15 May 2015* (pronounced *the fifteenth of May twenty fifteen / two thousand (and) fifteen*). In both cases, the day may be spelled so that the ordinal numeral is overtly marked: *May 15th, 2015*. The pronunciation of the day-month-year pattern with the preposition *of* shows that the month modifies the day and not vice versa; we should treat the written date as it is pronounced, even though the preposition is not visible (cf. (de Marneffe et al., 2021, p. 285)). The month is a temporal nominal modifier, `nmod:tmod`.[9] Similar patterns occur in some other European languages, such as Spanish (*el quince de mayo*, lit. *the fifteen of May*).

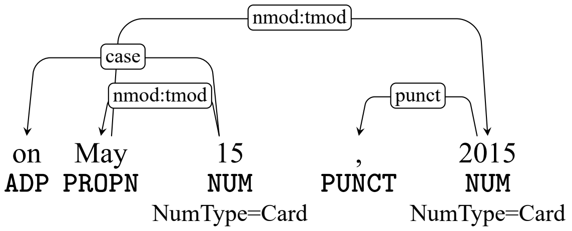
The situation is less clear with the month-day-year pattern. Schneider and Zeldes (2021) propose to make the day the head here, too, since the month can be omitted with sufficient context *(I'll see you on the fifteenth)*; if we wanted to omit the day instead, the case marker would have to change *(I'll see you **in** June)*. While this argumentation probably makes sense in English (no surface signals that the day modifies the month, parallel structure with the first pattern), note that the omission of the month could also be explained as ellipsis, and then the standard UD solution would be to promote the day to the head position (if the rule were that normally the month is the head).

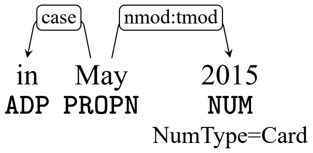
An analogous argument can be made about the year. It can be omitted from the full date *(on May 15(, 2015))* but it cannot occur with the day and without the month *(*on the fifteenth, 2015)*. We take this as evidence that the year modifies the month, rather than the day-month complex as a whole, and it should be attached to the month. Here we disagree with Schneider and Zeldes (2021), who attach the year to the day. In either case the relation is not `nummod` because the year is a label that does not express quantity (cf. (de Marneffe et al., 2021, p. 285)). It is a temporal nominal modifier, `nmod:tmod`. An optional era specifier *(BC/AD)* will be attached as `nmod:tmod` to the year.

Note that the year can appear with the month and without the day if the case marker is changed: *in May 2015*. Both *in May* and *in 2015* are grammatical; however, in the right context, *in May* has the same meaning as *in May 2015* while *in 2015* refers to a longer period. Therefore we propose to attach the year as a dependent of the month. Furthermore, it is also parallel to expressions where *of* is overtly used: *in October of 2002*.







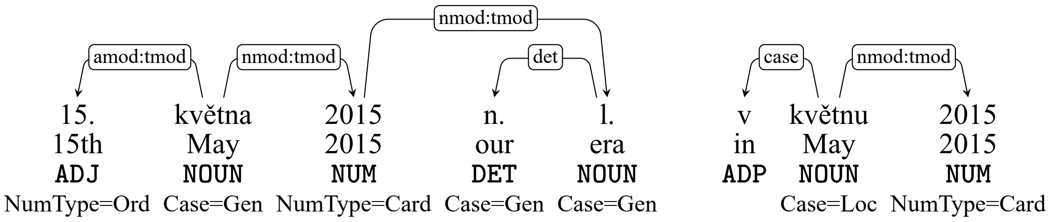


4.2 Czech

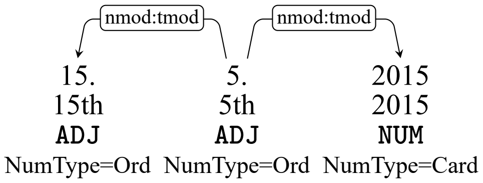
In Czech, the standard word order is day-month-year: *15. května 2015* (pronounced *patnáctého května dva tisíce patnáct*, lit. *fifteenth*.Gen *May*.Gen *2015*.Nom). The day is an ordinal numeral (`ADJ`) but unlike in English, it modifies the noun that denotes the month. This is semantically slightly odd (we are referring to the fifteenth day of May, not to a fifteenth May in a sequence of Mays), and it likely stems from a longer expression "the fifteenth day of May", but the morphosyntactic behavior has developed to that of regular

[9] The current practice in English UD seems to be that the `:tmod` subtype is only used when there is no preposition, as a kind of justification why there is `nmod` without case. We believe that it is equally useful to use it for prepositional temporal modifiers. Similarly, `advmod:tmod` could be used instead of plain `advmod` for time-related adverbs, as it is currently used in some other UD languages. However, regardless of how broadly they are applied, relation subtypes are always optional in UD.

adjectival modifiers. The day adjective thus agrees with the month in gender, number, and case. In the example above both of them have the genitive form, which is the default for temporal oblique modifiers, but both of them will switch to the nominative if the date is used as a subject, accusative if used as an object etc. Similar patterns occur in some other European languages, such as German (*am fünfzehnten Mai*, lit. *on-the*.Dat *fifteenth*.Dat *May*.Dat).[10]



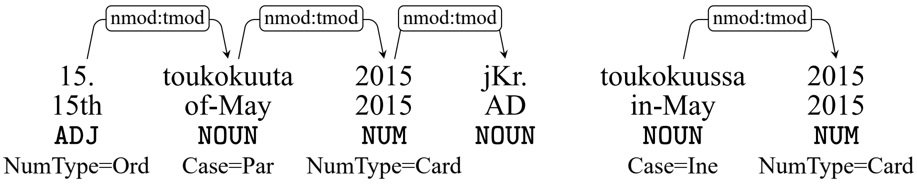
The above reasoning does not work so much when the month is encoded numerically: *15. 5. 2015*. It is conventionally pronounced *patnáctého pátý dva tisíce patnáct*, lit. *fifteenth*.Gen *fifth*.Nom *2015*.Nom, that is, there are two ordinals (plus the year cardinal) and they no longer agree in case. We do not have an explanation for this reading; nevertheless, in the absence of morphosyntactic evidence for one of the possible analyses, we propose to use a parallel structure to that of spelled-out month, i.e., the month is the head. The only change is the label of the first dependency, `nmod:tmod` instead of `amod:tmod`, as the relation does not behave like standard adjectival modification in Czech.



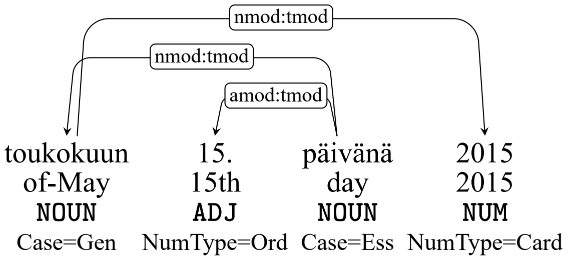
4.3 Finnish

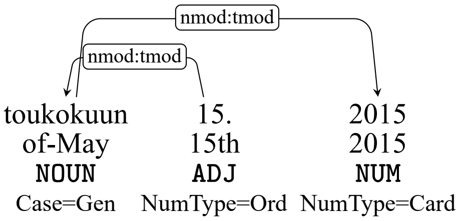
The most frequent date form in Finnish is *15. toukokuuta 2015*, pronounced *viidestoista toukokuuta kaksituhattaviisitoista*, lit. *fifteenth*.Nom *May*.Par *2015*.Nom. The day number is ordinal (`ADJ`) and it does not agree in case with the month name; instead, it forces the month name into the partitive form. We take this as a sign that the month depends on the day, like in English and unlike in Czech. The partitive case is used also when the month is spelled and pronounced as an ordinal number: *15.5.2015*, pronounced *viidestoista viidettä kaksituhattaviisitoista*. A month name with year (but without the day number) is typically found in the inessive case (*toukokuussa 2015* "in May 2015") but can occur in other cases if they are required by the surrounding syntactic context. Their relation should be `nmod:tmod` because the number does not specify a quantity of months.

Finnish also has an alternative date pattern, common in informal and spoken situations, where the month precedes the day. Here the month must be in the genitive case and the day takes the essive form. The day ordinal is optionally followed by the word "day", also in essive: *toukokuun viidentenätoista (päivänä)*. Another option is that the day ordinal and the word "day" are in nominative: *toukokuun viidestoista päivä*. Here the day ordinal modifies *päivä* and agrees with it in case.

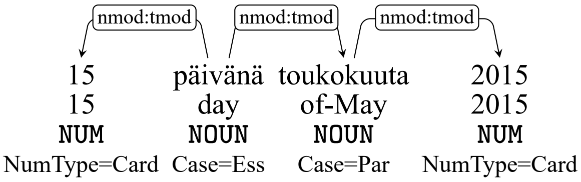


[10] There are also German examples where the month is in the genitive and modifies the dative day ordinal: *am Fünfzehnten des Monats* "on the fifteenth of the month".


nmod:tmod
nmod:tmod
amod:tmod
toukokuun 15. päivänä 2015
of-May 15th day 2015
NOUN ADJ NOUN NUM
Case=Gen NumType=Ord Case=Ess NumType=Card

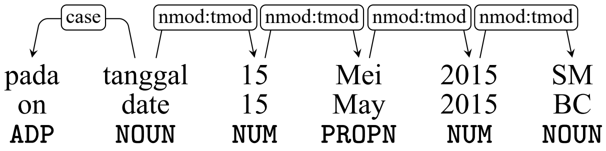


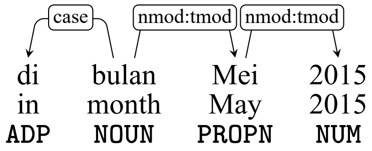
Finally, it is also possible to encounter date expressions that feature the word *päivä* "day" but the preceding number is not properly marked as ordinal (the period is missing): *15 päivänä toukokuuta 2015*. We acknowledge that it may then be tagged as a cardinal, although it may be pronounced as an ordinal (*viidentenätoista* "fifteenth.Ess" instead of *viitenätoista* "fifteen.Ess"). The relation from *päivänä* can be `nmod:tmod` but not `nummod`.


nmod:tmod
nmod:tmod
nmod:tmod
15 päivänä toukokuuta 2015
15 day of-May 2015
NUM NOUN NOUN NUM
NumType=Card Case=Ess Case=Par NumType=Card

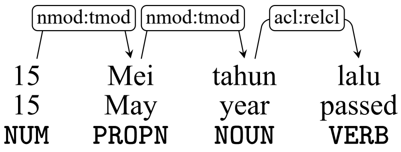
4.4 Indonesian

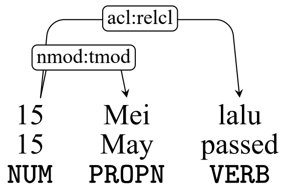
In Indonesian, dates are often introduced by the word *tanggal* "date": *pada tanggal 15 Mei 2015*, pronounced *pada tanggal lima belas Mei dua ribu lima belas*, lit. *on date five -teen May two thousand five -teen*. The day and year numbers are cardinal (rather than ordinal) numerals. As always, they do not denote quantity of anything, so they should not be attached via `nummod`. There are no morphological clues that would enlighten the relation between the day and the month, so a `flat` analysis seems appealing. Nevertheless, we can at least repeat what Schneider and Zeldes (2021) said about English. The month can be omitted and we can say *pada tanggal lima belas* "on the fifteenth". If that leads to making the day the head in English, we should make the Indonesian analysis parallel and attach the month to the day as `nmod:tmod`. The year, if present, will be attached to the month, and the optional era specifier modifies the year.

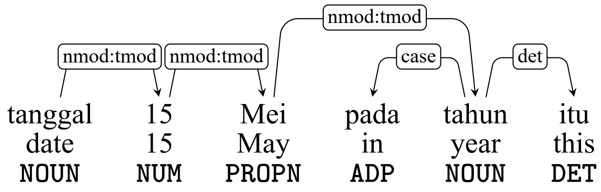




Modifiers such as *lalu* "last" and *mendatang* "next" are tagged `ADJ` in some treebanks and `VERB` in others (*last = passed, next = coming*). Of course, this should be harmonized, but the adjective-verb distinction is beyond the scope of this paper. If the modifier is a verb, it is a relative adnominal clause ("that has passed") and should be attached as `acl:relcl`. Otherwise it is a simple adjectival modifier, `amod`. In *tahun lalu* "last year", *lalu* modifies *tahun*. We could also say *15 Mei lalu* "last May 15", which may or may not be in the previous year, and it is typically both the fifteenth day of the last May, and the last fifteenth of May. Hence both the day and the month could serve as the parent node, and in the absence of other criteria, we propose to attach the modifier high:



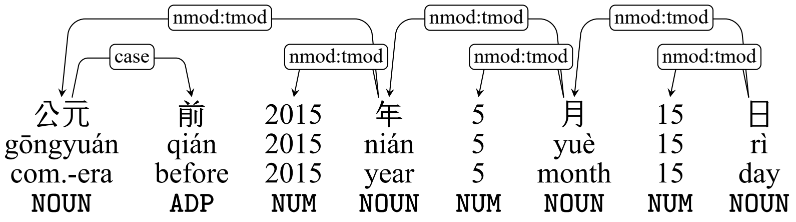


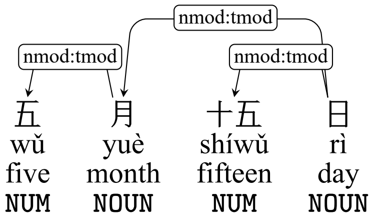


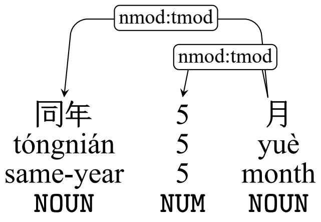
4.5 Chinese

In Chinese, dates proceed from the least specific to the most specific item. Numbers are always accompanied by the nouns for “year”, “month” and “day”; there are no names for months, they are encoded simply as a number + “month”. Years are normally written using Western Arabic digits. Months and days either use Arabic digits, too, or they are written in Chinese characters. Examples: 2015年5月15日 (*2015 nián 5 yuè 15 rì*, pronounced *èr líng yī wǔ nián wǔ yuè shíwǔ rì*), lit. *two zero one five year five month fifteen day*; 五月十五日 *(wǔ yuè shíwǔ rì)* “May 15”.

The numerals are cardinal and modify the respective nouns, but their relation should be `nmod:tmod` rather than `nummod`, as they do not denote quantity. Similarly to Indonesian, there are hardly any criteria that would favor one of the three items as the head. We therefore propose an analysis that is parallel to Indonesian and English, i.e., the less specific item depends on the more specific one. An era specifier, if present, modifies the year expression and is attached to its head noun. The year number may be substituted by an expression such as 同年 *(tóngnián)* “same year”, 次年 *(cìnián)* “following year” etc.

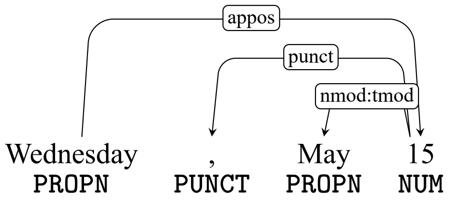

nmod:tmod
case
nmod:tmod
nmod:tmod
nmod:tmod
nmod:tmod
nmod:tmod
公元 前 2015 年 5 月 15 日
gōngyuán qián 2015 nián 5 yuè 15 rì
com.-era before 2015 year 5 month 15 day
NOUN ADP NUM NOUN NUM NOUN NUM NOUN

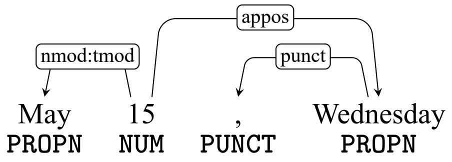




5 Days of Week

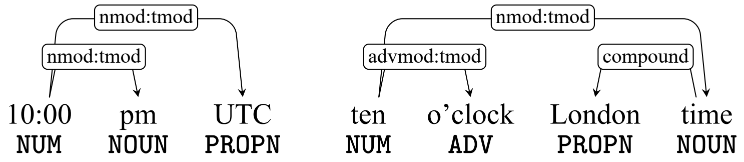
When the name of the day of week occurs together with the date, it can be understood as an apposition. Both expressions refer to the same day and they can be reordered. The first expression is treated as the technical head.



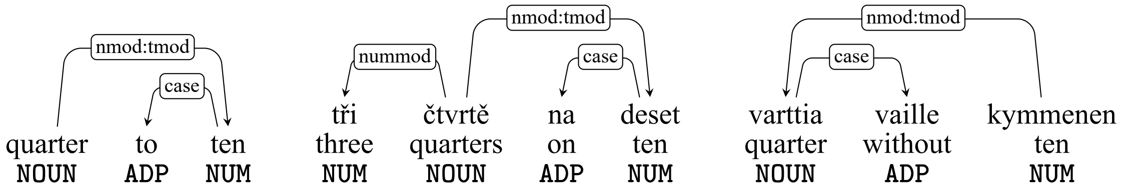


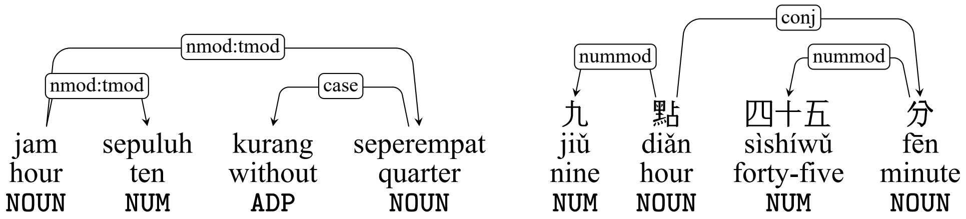
6 Time

We propose in Section 2 that time expressed using digits be one token, as in the following example (copied from Schneider and Zeldes (2021)). Schneider and Zeldes also propose that if the time is written as *ten o’clock*, the token *o’clock* should be considered an adverb and `advmod` of *ten*.



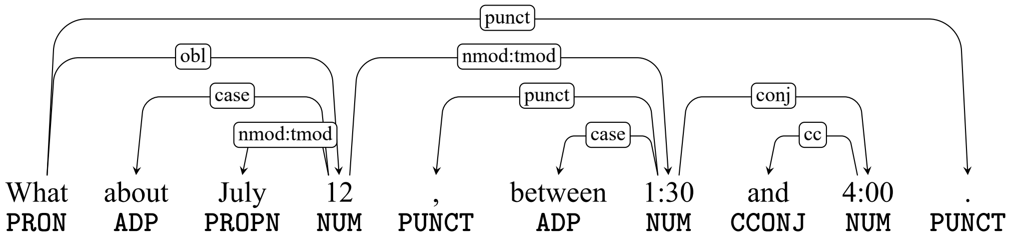
In general, verbose time expressions (as opposed to numbers) can vary substantially across languages, resulting in different analyses.





Note that in English, Czech, Finnish and Indonesian, the prepositions indicate which part is the modifier. In Chinese, we have two numbers with units. Unlike in dates, the `nummod` analysis is quite appropriate here, as we are counting hours (and minutes). This is not the case in Indonesian, where *sepuluh jam* "ten hours" is `nummod` (indicating duration) but *jam sepuluh* (lit. *the hour ten*) is `nmod:tmod` (labeling the hour). The relation between the Chinese hours and minutes cannot be characterized as subordination, hence we propose `conj` instead of `nmod:tmod` (*nine hours [and] forty-five minutes*; another possible candidate would be `flat`).

If time occurs together with date *(it happened on May 15 at 9:45)*, they are often two independent modifiers — note that in the previous English example, each has its own preposition. However, there are situations where date and time have to be considered as one unit, denoting a point in time, which has a syntactic function in the sentence: *What about July 12, between 1:30 and 4:00.* It is not obvious whether the date or the time should serve as the head in such cases. In this particular English example, we could say that the preposition *about* belongs to the date, hence the following analysis:

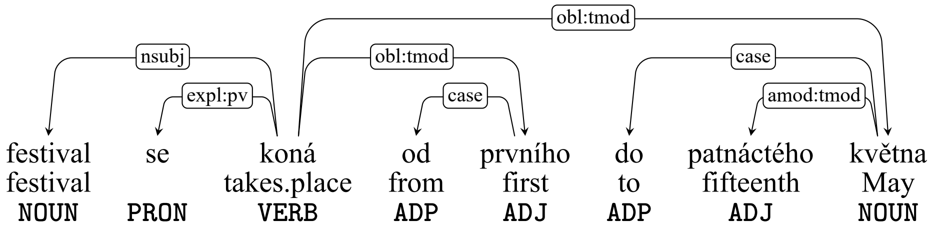


7 Ranges

Dates and times often come in ranges, as in the sentence *The festival takes place from May 15 to June 10.* In writing, the range can be signaled by a dash *(It takes place May 15 – June 10.)* The range can occur at various levels of precision, e.g. *It takes place May 15 – 16*, or *It takes place from May to November 2015*.

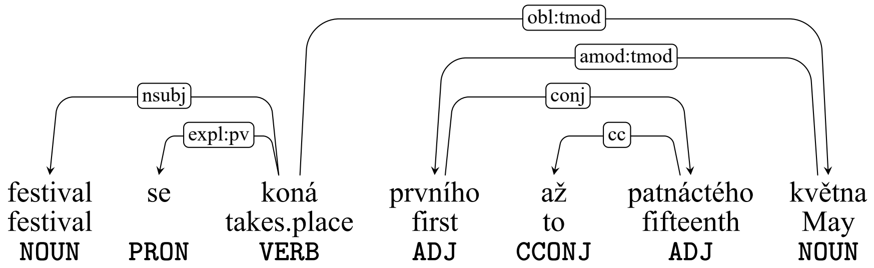
There are several options how to analyze ranges. Two full date expressions with different prepositions (*from* and *to*) could be quite naturally annotated as two sibling oblique modifiers of the same clause. Cases where only a part of the date is ranged (e.g., the month in *from May to November 2015*) could be handled as ellipsis (i.e., the first modifier would be *from May* and the second would be *to November 2015*). Analogously with a dash, the first expression would be *May* and the second – *November 2015*.

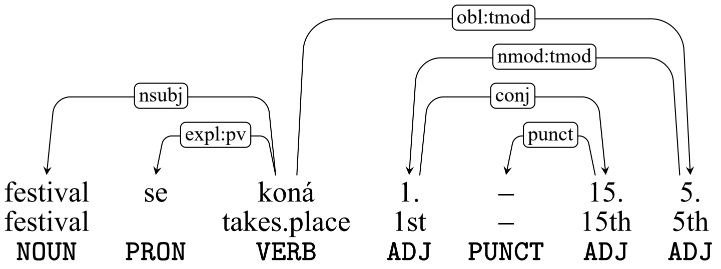
This solution has the disadvantage that the partial first expression is detached from the shared part, so it is more difficult to infer that *May* actually refers to *May 2015*.



Another option is to attach the closing part of the range to the opening part; for partial ranges, only the ranged parts can be connected, and the shared, less specific part is attached to the head. This approach is currently taken in some UD treebanks;[11] however, if the closing part has a preposition (either spelled out, or assumed to be encoded by the dash), the annotators mechanically pick the nmod relation, which seems wrong. The second date does not really modify the first date. Their relation is much closer to coordination: the event occurs on both the dates (as well as on all dates in between). In fact, some languages use conjunctions instead of (or in addition to) prepositions to express ranges: German *Mai bis November*, Czech *květen až listopad* "May to November". Treating the prepositional *(from–to)* cases as coordination would have the advantage of better parallelism and it would solve the shared modifier problem (*from May to November 2015* would be analogous to *either May or November 2015*). Coordinating May with November would allow *2015* to technically modify the first conjunct but be propagated in enhanced UD to the second conjunct.[12]

We lean towards the coordination analysis as the most parallel and cross-linguistically applicable solution. If that is not accepted by the UD community, then we think that attaching the two endpoints as siblings is better than making one of them an nmod of the other.





8 How to Fix the Treebanks

We provide a survey of annotation patterns in the UD 2.8 treebanks of English, Czech, Finnish, Indonesian and Chinese in the appendices. Some treebanks are internally consistent, some less so, but there is very little consensus across treebanks of the same language. It is thus obvious that any improvement would be welcome, even if it cannot be done perfectly.

[11] In fact, this approach also matches example (49) of (de Marneffe et al., 2021).

[12] As one of the reviewers pointed out, the correlative expression *from X to Y* is certainly a grammaticalized construction (it cannot be paraphrased as *to November from May*), and the constituents need not be nominals (*Heights range from tall to short*, etc.), reminiscent of coordination. See (Reynolds and Pullum, 2013) for an argument that *versus* has grammaticalized from a preposition to a coordinator.

Fortunately, it is not necessary to re-annotate all UD treebanks manually. Language-specific patterns can be designed that will find (almost) all occurrences of date and time expressions in a treebank,[13] and identify their parts. The annotation can then be harmonized using tree-rewriting systems such as Udapi (Popel et al., 2017) or Grew (Guillaume, 2021).

9 Conclusion

We have surveyed various date/time-related expressions in five languages from four different language families. We have shown that some of these expressions in some languages have internal morphosyntactic structure, which should be observed when constructing their UD analysis. The syntactic structure of semantically corresponding expressions is not always compatible across language boundaries, hence the annotation rules cannot be language-independent. However, in cases where no underlying syntax can be detected, we recommend one of the annotation options as the default solution.

We believe that it is necessary to add some guidelines for date and time expressions to the UD documentation, as it will greatly improve consistency of the UD data (these expressions are quite frequent in some genres). We also believe that with language-specific heuristics, the data can be fixed relatively easily, using existing tools for automatic modification of the dependency structures.

Acknowledgements

This work was supported by the grants 20-16819X (LUSyD) of the Czech Science Foundation; and LM2018101 (LINDAT/CLARIAH-CZ) of the Ministry of Education, Youth, and Sports of the Czech Republic.

We thank the anonymous reviewers for very useful comments.

References

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

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.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia. European Language Resources Association.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, pages 4034–4043, Marseille, France. European Language Resources Association.

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, Göteborg, Sweden.

Brett Reynolds and Geoffrey K. Pullum. 2013. New members of 'closed classes' in English. 02.

Nathan Schneider and Amir Zeldes. 2021. Mischievous nominal constructions in Universal Dependencies. In *Proceedings of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*, Sofia, Bulgaria.

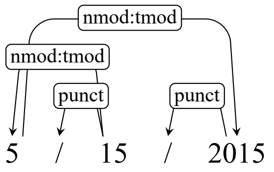
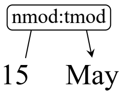
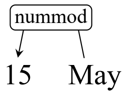
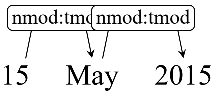
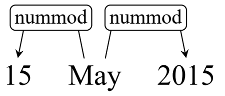
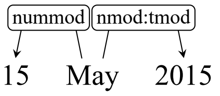
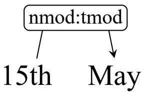
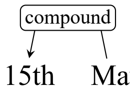
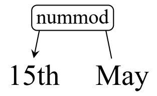
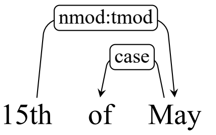
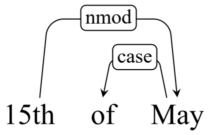
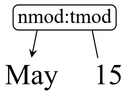
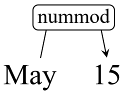
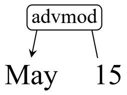
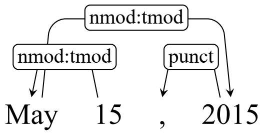
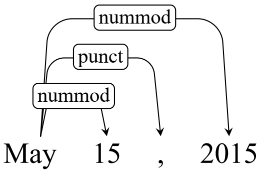
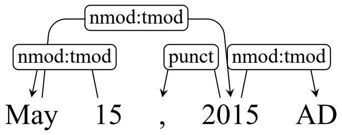
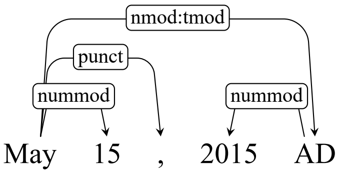
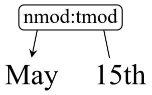
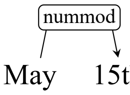
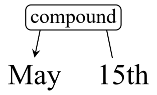
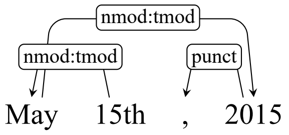
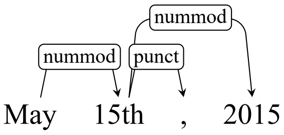
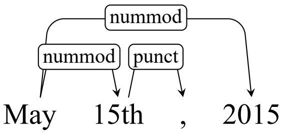
[13] Lists of language-specific names for months and week days are very useful in such heuristics.

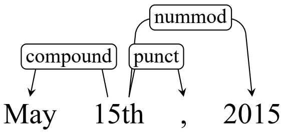
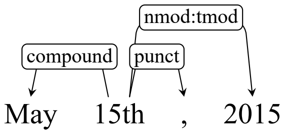
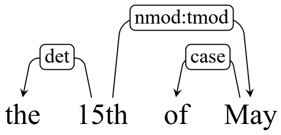
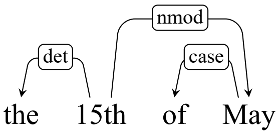
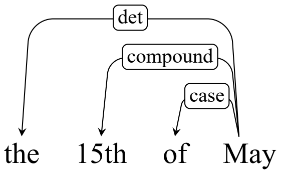
A Survey of English Date Annotations in UD 2.8

We provide an overview of date annotations currently applied in the English UD treebanks. The survey demonstrates the variability of dependency relations. There are further differences in POS tags and features but they are not shown here. We only show dates, i.e., without days of week, without times, no ranges etc. Sometimes the same pattern receives multiple annotations within the same treebank. Other treebanks are more consistent internally, yet the approaches differ heavily across treebanks.

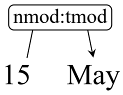
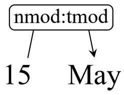
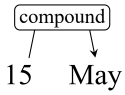
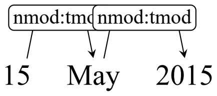
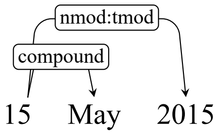
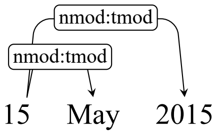
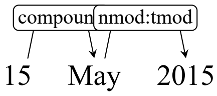
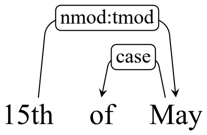
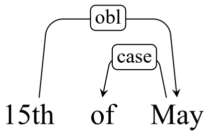
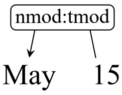
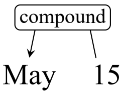
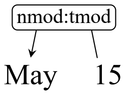
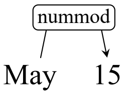
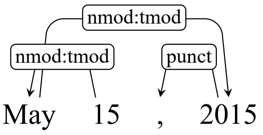
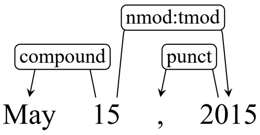
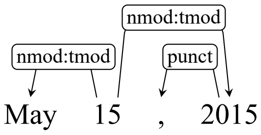
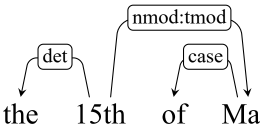
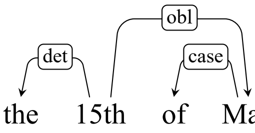
The total numbers of occurrences in the tables are approximate. Some marginal cases with extra dependents are either clustered with more generic patterns or omitted from the table completely.

A.1 EWT

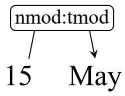
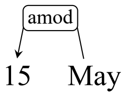
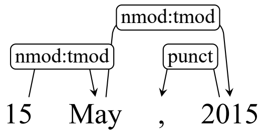
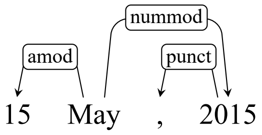
Pattern	Proposal	Total	EWT Trees
5/15/2015		183	5/15/2015 (one token)
15 May		11	
15 May 2015		20	 
15th May		3	 
15th of May		2	
May 15		80	 
May 15, 2015		45	
May 15, 2015 AD	 nmod:tmod nmod:tmod punct nmod:tmod May 15 , 2015 AD	2	
May 15th		23	 
May 15th, 2015		5	 

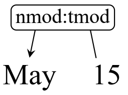
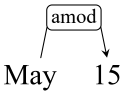
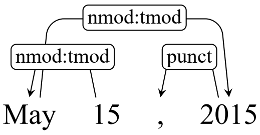
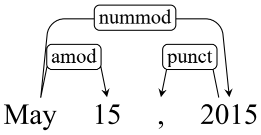
			 
the 15th of May		3	 

A.2 GUM

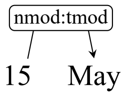
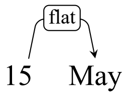
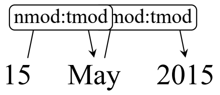
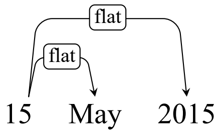
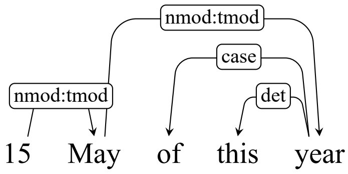
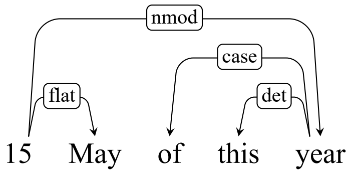
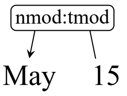
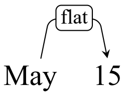
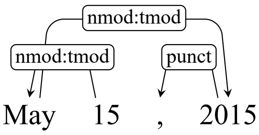
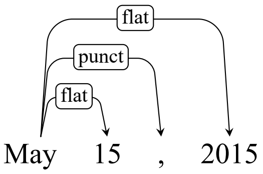
Pattern	Proposal	Total	GUM Trees
15 May		4	 
15 May 2015		16	  
15th of May		1	
May 15		21	  
May 15, 2015		82	 
the 15th of May		1	

A.3 LinES

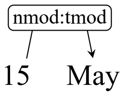
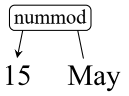
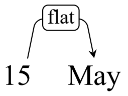
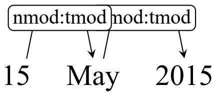
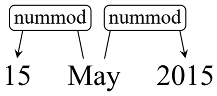
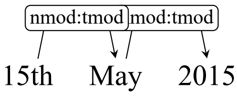
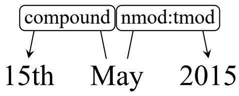
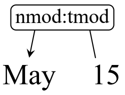
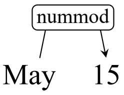
Pattern	Proposal	Total	LinES Trees
15 May		4	
15 May, 2015		2	

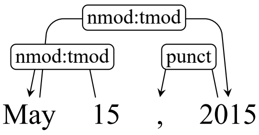
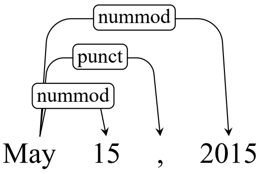
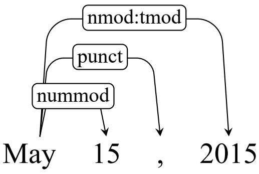
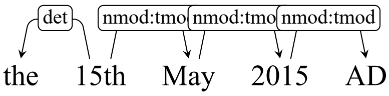
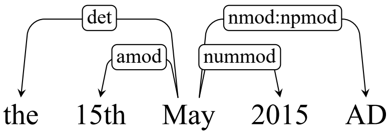
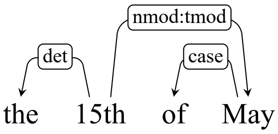
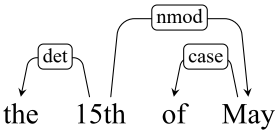
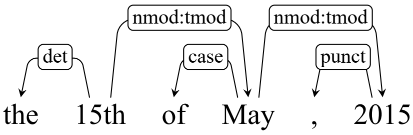
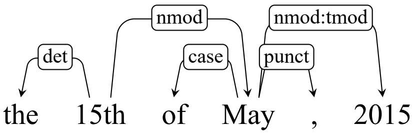
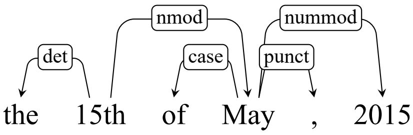
May 15		1	
May 15, 2015		1	

A.4 ParTUT

Pattern	Proposal	Total	ParTUT Trees
15 May		5	
15 May 2015		50	
15 May of this year	 nmod:tmod case nmod:tmod det 15 May of this year	1	 nmod case flat det 15 May of this year
May 15		1	
May 15, 2015		5	

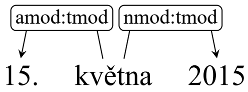
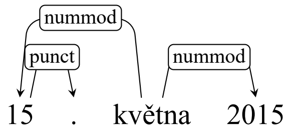
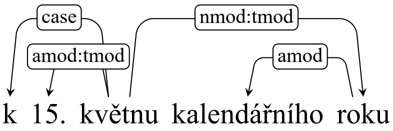
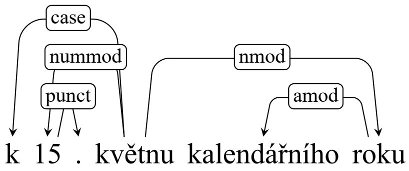
A.5 PUD

Pattern	Proposal	Total	PUD Trees
15 May		3	 
15 May 2015		4	
15th May 2015		1	
May 15		1	

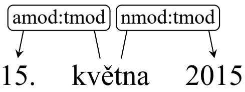
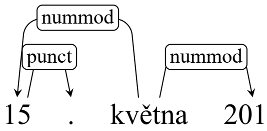
May 15, 2015		5	 
the 15th May 2015 AD	 det nmod:tmo nmod:tmo nmod:tmod the 15th May 2015 AD	1	
the 15th of May		4	
the 15th of May, 2015	 nmod:tmod nmod:tmod det case punct the 15th of May , 2015	3	 nmod nmod:tmod det case punct the 15th of May , 2015  nmod nummod det case punct the 15th of May , 2015

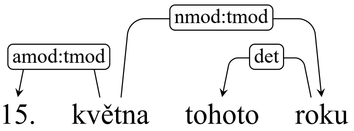
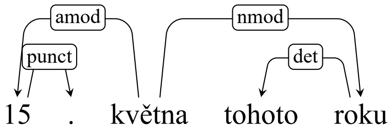
B Survey of Czech Date Annotations in UD 2.8

B.1 CLTT

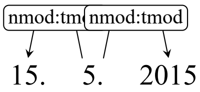
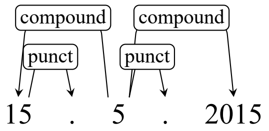
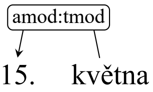
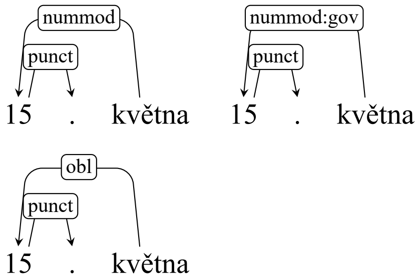
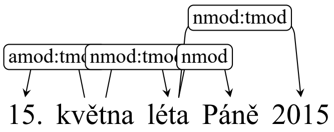
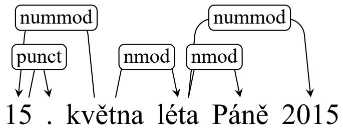
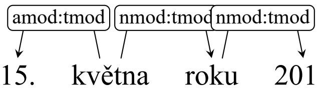
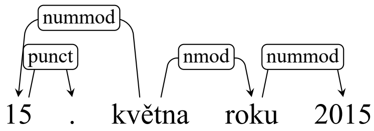
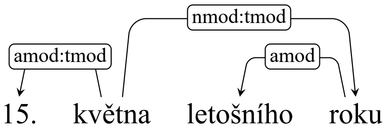
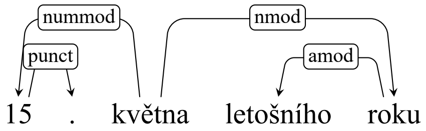
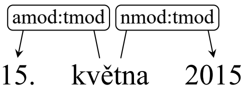
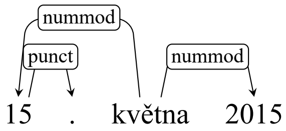
Pattern	Proposal	Total	CLTT Trees
15. května 2015		12	
k 15. květnu kalendářního roku	 case nmod:tmod amod:tmod amod k 15. květnu kalendářního roku	1	 case nummod nmod punct amod k 15 . květnu kalendářního roku

B.2 FicTree

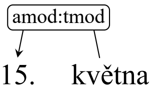
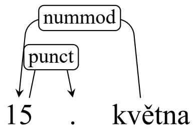
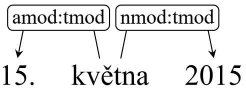
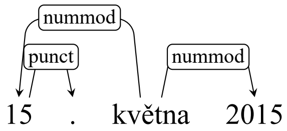
Pattern	Proposal	Total	FicTree Trees
15. května 2015		5	

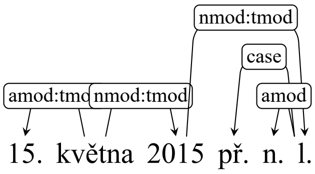
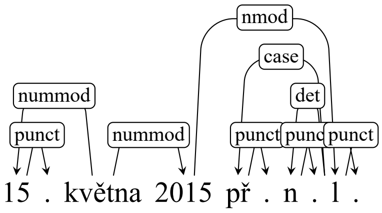
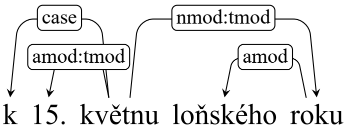
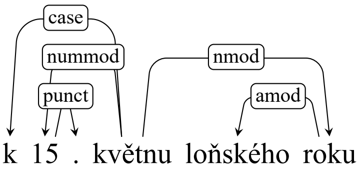
15. května tohoto roku		1	

B.3 PDT

Pattern	Proposal	Total	PDT Trees
15. 5. 2015		126	
15. května		782	 nummod nummod:gov punct punct 15 . května 15 . května obl punct 15 . května
15. května léta Páně 2015	 nmod:tmod amod:tm nmod:tmo nmod 15. května léta Páně 2015	1	 nummod nummod punct nmod nmod 15 . května léta Páně 2015
15. května roku 2015		1	
15. května letošního roku		42	 nummod nmod punct amod 15 . května letošního roku
15. května 2015		240	

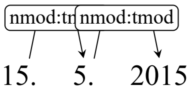
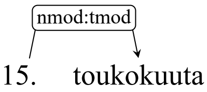
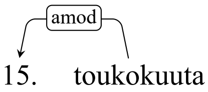
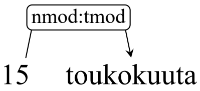
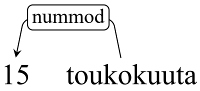
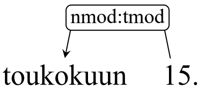
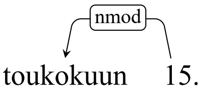
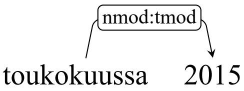
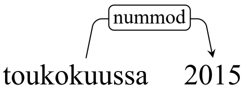
B.4 PUD

Pattern	Proposal	Total	PUD Trees
15. května		8	
15. května 2015		10	

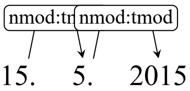
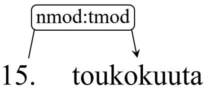
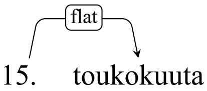
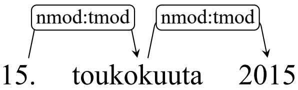
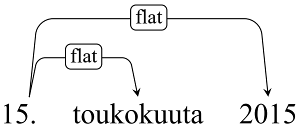
15. května 2015 př. n. l.	 nmod:tmod case amod:tmo nmod:tmod amod 15. května 2015 př. n. l.	1	 nmod case nummod det punct nummod punct punc punct 15 . května 2015 př . n . l .
k 15. květnu loňského roku	 case nmod:tmod amod:tmod amod k 15. květnu loňského roku	1	 case nummod nmod punct amod k 15 . květnu loňského roku

C Survey of Finnish Date Annotations in UD 2.8

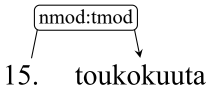
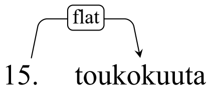
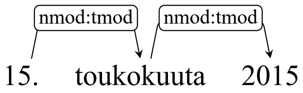
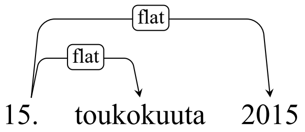
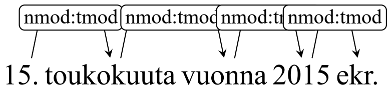
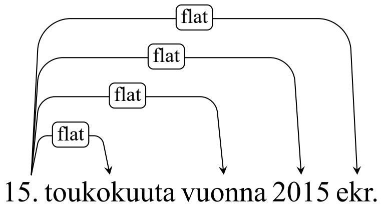
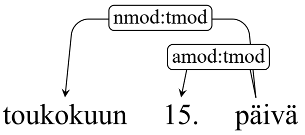
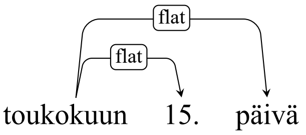
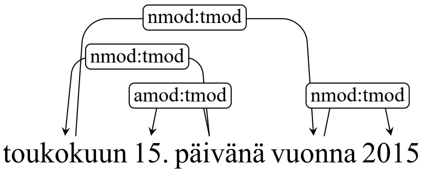
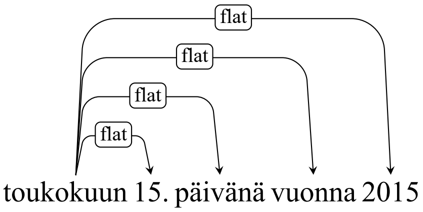
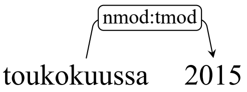
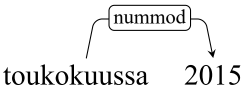
C.1 FTB

Pattern	Proposal	Total	FTB Trees
15.5.2015		10	15.5.2015
15. toukokuuta		12	
15 toukokuuta		1	
toukokuun 15.		1	
toukokuussa 2015		4	

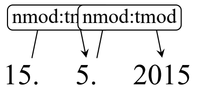
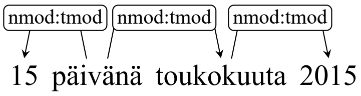
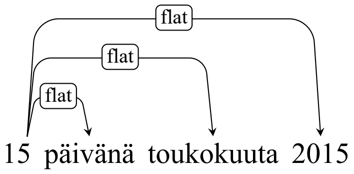
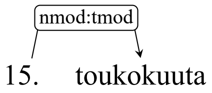
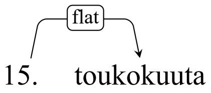
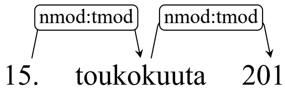
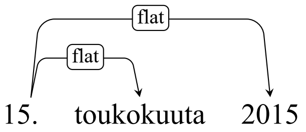
C.2 OOD

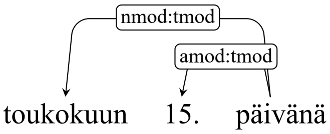
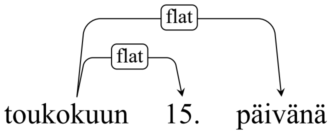
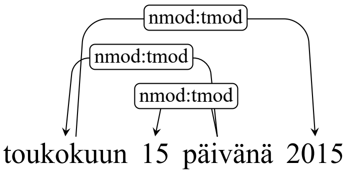
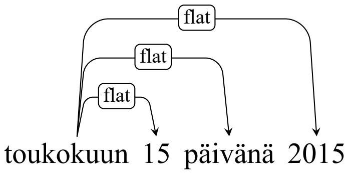
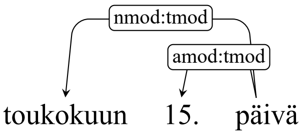
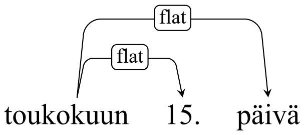
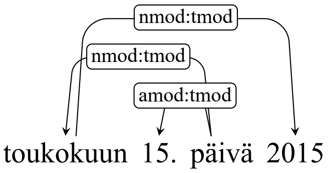
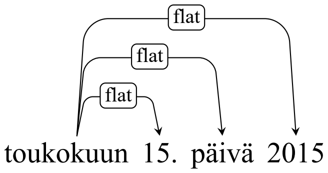
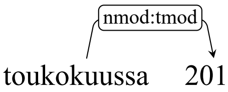
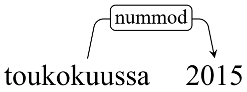
Pattern	Proposal	Total	OOD Trees
15.5.2015		18	15.5.2015
15. toukokuuta		3	
15. toukokuuta 2015		3	

C.3 PUD

Pattern	Proposal	Total	PUD Trees
15. toukokuuta		4	
15. toukokuuta 2015		12	
15. toukokuuta vuonna 2015 ekr.	 nmod:tmod nmod:tmo nmod:t nmod:tmod 15. toukokuuta vuonna 2015 ekr.	1	 flat flat flat flat 15. toukokuuta vuonna 2015 ekr.
toukokuun 15. päivä		1	
toukokuun 15. päivänä vuonna 2015	 nmod:tmod nmod:tmod amod:tmod nmod:tmod toukokuun 15. päivänä vuonna 2015	1	 flat flat flat flat toukokuun 15. päivänä vuonna 2015
toukokuussa 2015		15	

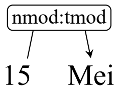
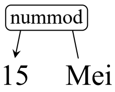
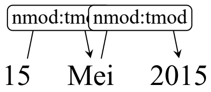
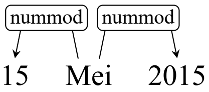
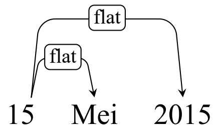
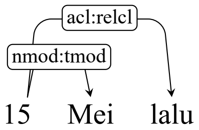
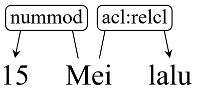
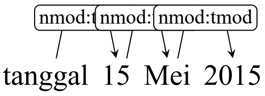
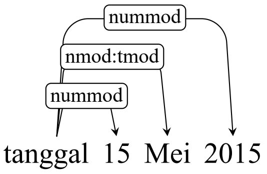
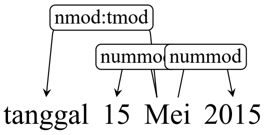
C.4 TDT

Pattern	Proposal	Total	TDT Trees
15.5.2015		23	15.5.2015
15 päivänä toukokuuta 2015	 nmod:tmod nmod:tmod nmod:tmod 15 päivänä toukokuuta 2015	120	
15. toukokuuta		47	
15. toukokuuta 2015		185	

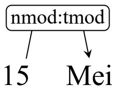
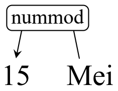
toukokuun 15. päivänä		1	
toukokuun 15 päivänä 2015	 nmod:tmod nmod:tmod nmod:tmod toukokuun 15 päivänä 2015	1	
toukokuun 15. päivä		1	
toukokuun 15. päivä 2015	 nmod:tmod nmod:tmod amod:tmod toukokuun 15. päivä 2015	2	
toukokuussa 2015		50	

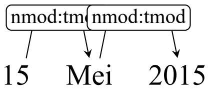
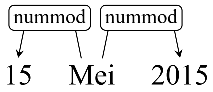
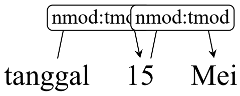
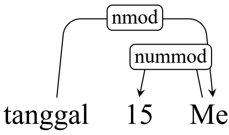
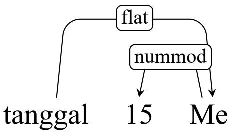
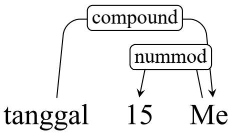
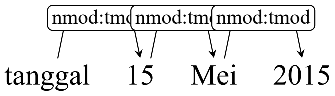
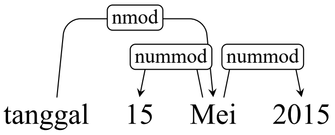
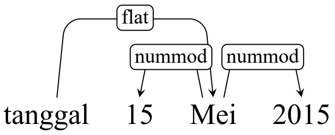
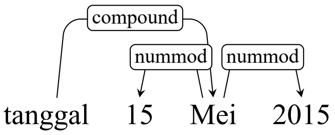
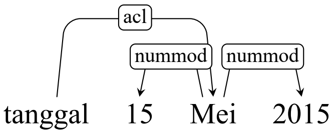
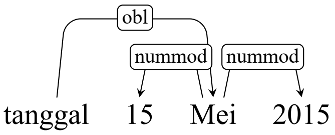
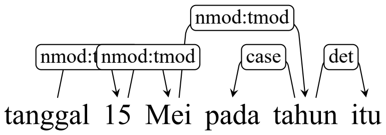
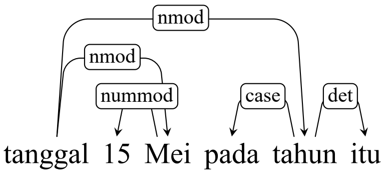
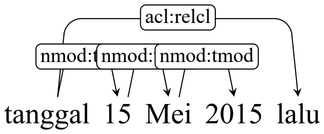
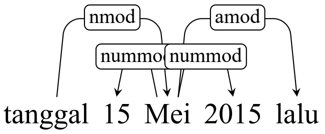
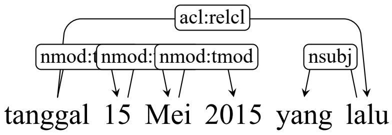
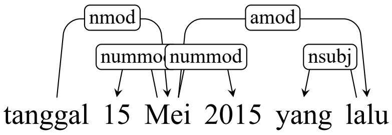
D Survey of Indonesian Date Annotations in UD 2.8

D.1 CSUI

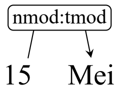
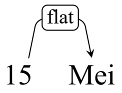
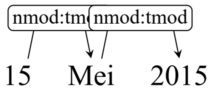
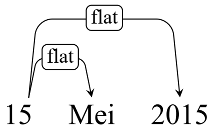
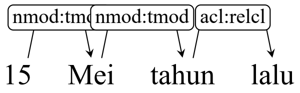
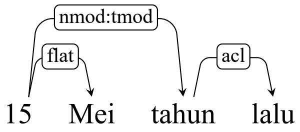
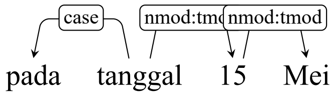
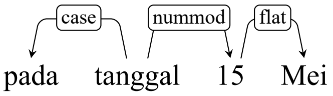
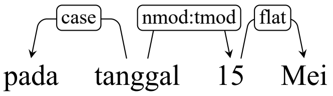
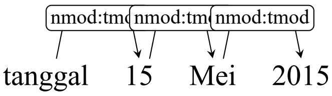
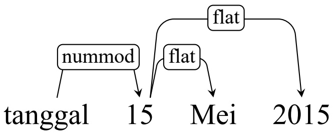
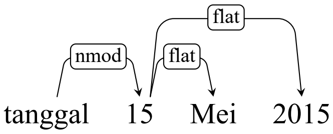
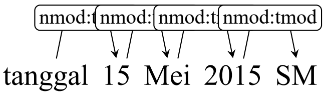
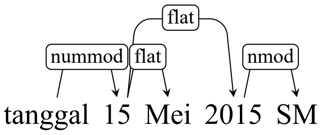
Pattern	Proposal	Total	CSUI Trees
15 Mei		12	
15 Mei 2015		32	 
15 Mei lalu		5	
tanggal 15 Mei 2015		5	 

D.2 GSD

Pattern	Proposal	Total	GSD Trees
15 Mei		22	

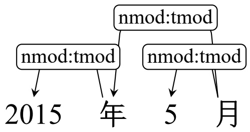
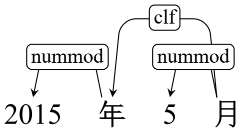
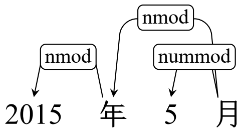
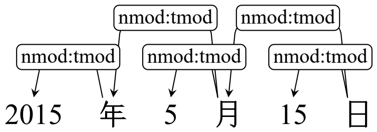
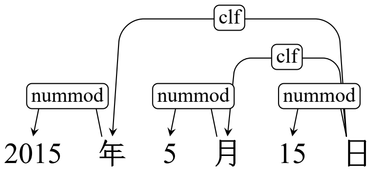
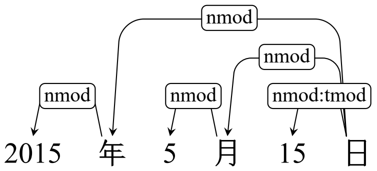
15 Mei 2015		153	
tanggal 15 Mei		9	  
tanggal 15 Mei 2015		100	    
tanggal 15 Mei pada tahun itu	 nmod:tmod nmod:t nmod:tmod case det tanggal 15 Mei pada tahun itu	1	 nmod nmod nummod case det tanggal 15 Mei pada tahun itu
tanggal 15 Mei 2015 lalu	 acl:relcl nmod:t nmod: nmod:tmod tanggal 15 Mei 2015 lalu	1	
tanggal 15 Mei 2015 yang lalu	 acl:relcl nmod:t nmod: nmod:tmod nsubj tanggal 15 Mei 2015 yang lalu	1	 nmod amod nummo nummod nsubj tanggal 15 Mei 2015 yang lalu

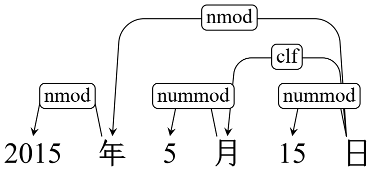
D.3 PUD

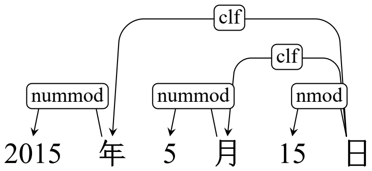
Pattern	Proposal	Total	PUD Trees
15 Mei		4	
15 Mei 2015		6	
15 Mei tahun lalu	 nmod:tm nmod:tmod acl:relcl 15 Mei tahun lalu	1	
pada tanggal 15 Mei	 case nmod:tm nmod:tmod pada tanggal 15 Mei	4	 
tanggal 15 Mei 2015	 nmod:tm nmod:tm nmod:tmod tanggal 15 Mei 2015	5	 
tanggal 15 Mei 2015 SM	 nmod:t nmod: nmod:t nmod:tmod tanggal 15 Mei 2015 SM	1	 flat nummod flat nmod tanggal 15 Mei 2015 SM

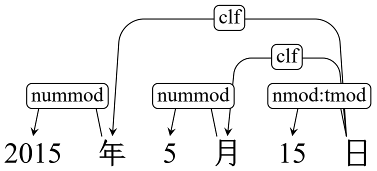
E Survey of Chinese Date Annotations in UD 2.8

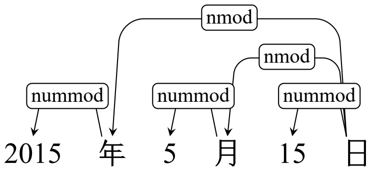
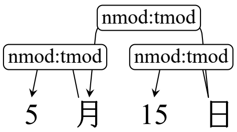
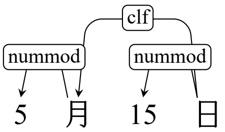
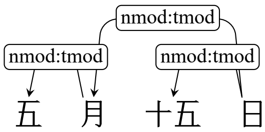
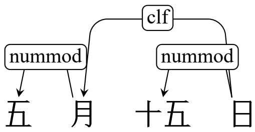
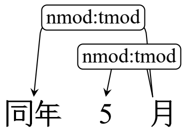
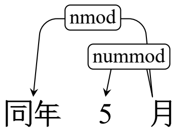
E.1 GSD

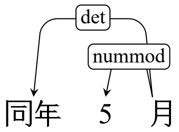
Pattern	Proposal	Total	GSD Trees
2015年5月	 nmod:tmod nmod:tmod nmod:tmod 2015 年 5 月	141	 
2015年5月15日	 nmod:tmod nmod:tmod nmod:tmod nmod:tmod nmod:tmod 2015 年 5 月 15 日	264	 clf clf nummod nummod nummod 2015 年 5 月 15 日  nmod nmod nmod nmod nmod:tmod 2015 年 5 月 15 日

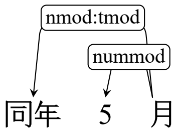
<table>
<tr><td></td><td></td><td></td><td>



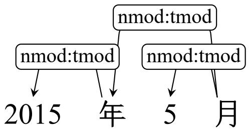
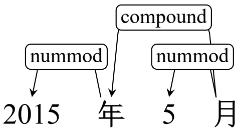
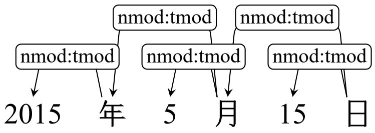
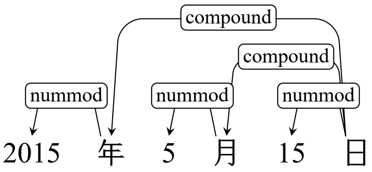


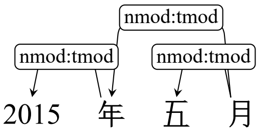
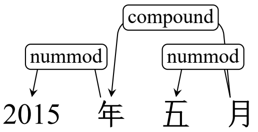
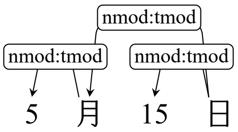
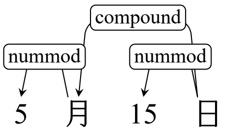
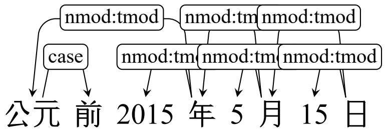
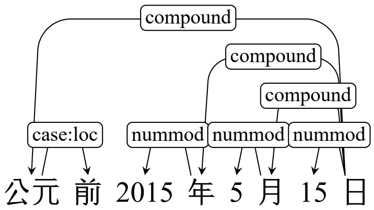
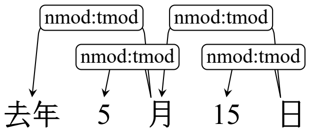
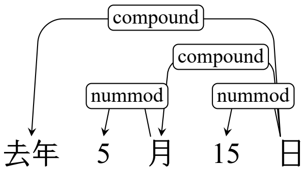

</td></tr>
<tr><td>5月15日</td><td>
</td><td>67</td><td>
</td></tr>
<tr><td>五月十五日</td><td>
</td><td>1</td><td>
</td></tr>
<tr><td>同年5月</td><td>
</td><td>21</td><td>




</td></tr>
</table>

E.2 PUD

<table>
<tr><th>Pattern</th><th>Proposal</th><th>Total</th><th>PUD Trees</th></tr>
<tr><td>2015年5月</td><td>
</td><td>14</td><td>
</td></tr>
<tr><td>2015年5月15日</td><td>
</td><td>12</td><td>
</td></tr>
</table>

2015年五月		1	
5月15日		6	
公元前2015年5月15日	 nmod:tmod nmod:tm nmod:tmod case nmod:tm nmod:tm nmod:tmod 公元 前 2015 年 5 月 15 日	1	 compound compound compound case:loc nummod nummod nummod 公元 前 2015 年 5 月 15 日
去年5月15日	 nmod:tmod nmod:tmod nmod:tmod nmod:tmod 去年 5 月 15 日	1	 compound compound nummod nummod 去年 5 月 15 日