

**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

HABILITATION THESIS

Daniel Zeman

**Cross-Language Harmonization
of Linguistic Resources**

Institute of Formal and Applied Linguistics (ÚFAL)

Prague 2023

Title: Cross-Language Harmonization of Linguistic Resources

Author: Daniel Zeman

Institute: Institute of Formal and Applied Linguistics (ÚFAL)

Abstract: The presented work consists of two parts. In the first part I summarize the main directions of my research since the defense of my PhD thesis in 2005. I start with cross-language transfer of parsing models to languages that have little or no annotated data. This section provides motivation for the subsequent sections, which revolve around designing a description of natural language systems that could be used for any language, leading to data resources that are interoperable and comparable cross-linguistically. The harmonization efforts culminate in the international project called Universal Dependencies (UD), to which I have contributed significantly. Finally, I discuss some more recent spin-offs from Universal Dependencies, showing the current and future directions of my research work.

The second part contains a selection of my publications from the same period. Each publication is accompanied with a comment that puts it in context and assesses its long-term impact. The publications in the second part are directly related to the individual topics in the first part and I highlight these connections using cross-references in both ways.

Keywords: annotated corpora; morphology; dependency syntax; delexicalized parsing; shared task

Contents

Introduction	2
1 Low-resource Languages	5
1.1 Dependency Parsing	5
1.2 Delexicalized Parsing	6
1.3 Using Parallel Data	8
1.4 Evaluation	10
2 Harmonization of Morphological Annotation	12
2.1 Interset	12
2.2 ‘Google’ Universal POS Tags	16
2.3 Universal Dependencies	16
2.3.1 Layered Features	17
2.4 UniMorph	18
3 Harmonization of Syntactic Annotation	19
3.1 HamleDT	19
3.2 Stanford Dependencies	19
3.3 Universal Dependencies	21
4 Multilingual Shared Tasks	24
5 Future Directions	27
6 Selected Publications	31
6.1 Cross-Language Parser Adaptation between Related Languages	31
6.2 Reusable Tagset Conversion Using Tagset Drivers	40
6.3 HamleDT: Harmonized Multi-language Dependency Treebank	47
6.4 Universal Dependencies	85
6.5 CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies	140
6.6 Towards Deep Universal Dependencies	162
Bibliography	172
List of Figures	182
List of Tables	183

Introduction

This thesis focuses on linguistic resources for many different languages. More specifically, it focuses on corpora that are annotated both morphologically and syntactically; since syntactic structure is typically expressed as a rooted tree, such corpora are called *treebanks*. They are invaluable resources for the study of language systems and, more generally, for digital humanities. For several decades it was also assumed that morphosyntactic analysis is an essential first step towards any application that assumes computational understanding of natural language, including machine translation. This assumption has now been drastically reduced by the advances of deep learning models, which can be tuned for the end-user task and can capture morphology and syntax internally, without seeing corresponding human-made annotation; however, such models do not reveal how they arrived at the output they were asked for and, consequently, they do not bring much insight about the language itself. In contrast, some insight about the language system can be obtained if morphosyntactic analysis is taken as the target task and a model (*a parser*) is trained on a human-annotated treebank to predict the annotation for previously unseen data. (Note that deep learning still plays a role, now in solving the parsing task.) Furthermore, morphosyntactically parsed text is useful as input for heuristics solving downstream tasks whenever there is not enough training data in the given language annotated directly for those tasks.

Morphological annotation, as understood in the present thesis, consists of three main pieces of information: the lemma of a word, its part-of-speech (POS) category, and a set of morphological feature-value pairs that characterize the annotated word form within an inflectional (or derivational) paradigm. Not all treebanks separate the POS category and the features in the way we just did here; part of speech itself can be (and often is) viewed as another feature with a pre-defined set of possible values. Depending on the terminology used by individual authors, the lemma is then accompanied by a *POS tag* or a *morphological tag*, which is a more or less compact encoding of the feature-value pairs.

Tagsets come with different expectations about how much can and should be disambiguated by context. For example, the English word *can* is either a modal auxiliary (as in *I can give you a ride*), or a noun (as in *I have a can full of fruit*). We can also derive a verb from the noun (as in *How to can fruits*). The surface ambiguity between the first *can* and the other two is purely coincidental and we definitely want to disambiguate them in text. The second and third *can* are related, one is derived from the other, but we still want to distinguish them because the syntactic rules applying to nouns and verbs are not compatible [Zeman, 2018].

Many different standards have been proposed for morphological tagging. Some differences are differences between languages; but even within one language, tagsets vary substantially in their level of granularity and choice of phenomena to capture. Table 1 demonstrates this on the example of tags denoting adjectives.

The syntactic structure of a sentence can be annotated in various ways, depending on the underlying theory. Most frameworks represent the sentence hierarchically as a rooted directed tree. In the present thesis we focus on *dependency trees*, whose nodes correspond (mostly) to words, and edges connecting them are

Language	Tagset	Tag
English	Penn Treebank	JJ, JJR, JJS
Swedish	Mamba	AJ
Swedish	Stockholm-Umeå	JJ POS UTR SIN IND NOM JJ POS UTR SIN IND GEN JJ POS UTR SIN DEF NOM
		...
Czech	Prague Dependency Treebank	AAMS1----1A---- AAMS2----1A---- AAMS3----1A----
		...

Table 1: Morphological / POS tag examples for various languages. The tags for adjectives as defined in the Penn Treebank [Marcus et al., 1993], Mamba [Teleman, 1974, Nilsson et al., 2005], Stockholm-Umeå Corpus [Gustafson-Capková and Hartmann, 2006, p. 20–21], and the Prague Dependency Treebank (PDT) [Hajič et al., 2000]. The three PDT tags represent only a fraction; as many as 378 feature combinations are possible in a regular adjective paradigm. Stockholm-Umeå is less rich, but still it has many more tags than the three displayed here.

typed dependencies. Usage of such structures in linguistics dates back to the seminal work of Tesnière [1959], and a number of dependency-syntactic theories evolved since then; therefore, narrowing syntactic annotation to dependency trees itself does not ensure that there is a single set of annotation rules that everyone uses. To illustrate this, we show two annotations of the English sentence *I saw the man who loves you* in Figure 1, one following the annotation guidelines of the Prague Dependency Treebank (henceforth Prague Dependencies, PD) [Hajič et al., 2000], and the other following Stanford Dependencies (henceforth SD) [de Marneffe et al., 2014]. Topologically, the sentence receives in both frameworks identical structure, but the labels of the dependency relations differ. Nevertheless, the tree shapes may differ, too, as we demonstrate on the sentence *Bell, based in Los Angeles, makes electronic and building products* (Figure 2). Note that in this case SD does not even treat all words as nodes (the function words *in* and *and* are reflected as parts of the dependency relation types `prep_in` and `conj_and`, respectively, but they are not nodes).

Structure of the Thesis

The thesis consists of two parts. In the first part, I summarize the main directions of my research from 2006 to the present. I start in Chapter 1 with cross-language transfer of parsing models to languages with little or no annotated resources. This provides motivation for cross-linguistic harmonization of data resources, the topic of Chapter 2 (morphological harmonization) and Chapter 3 (syntactic harmonization). Chapter 4 returns to parsing and discusses several shared tasks that took advantage of harmonized data. Finally, Chapter 5 discusses some recent projects and future directions based on the work described in the previous

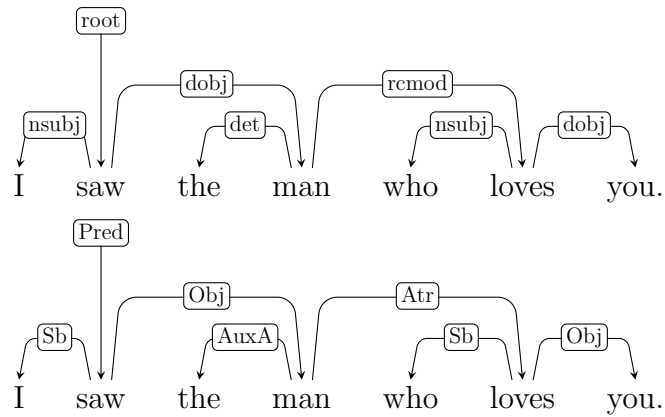


Figure 1: The sentence “*I saw the man who loves you*” in SD (up) and PD (down). Adapted from de Marneffe et al. [2006].

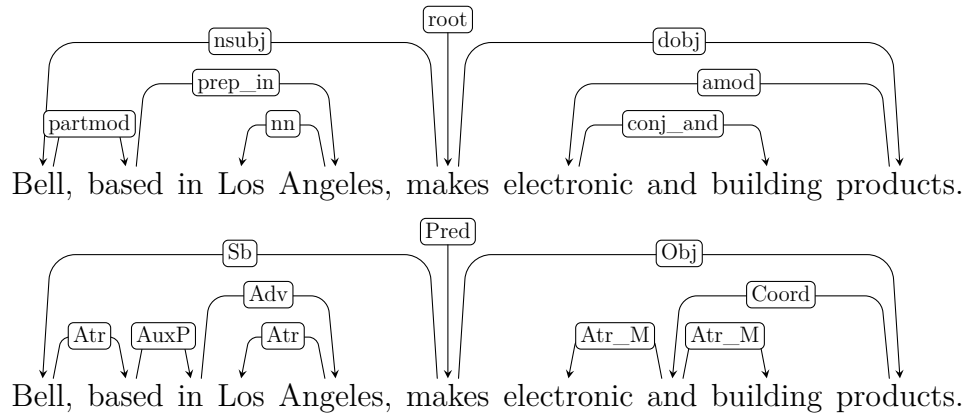


Figure 2: The sentence “*Bell, based in Los Angeles, makes electronic and building products*” in SD (up) and PD (down). Adapted from de Marneffe and Manning [2008].

chapters.

The second part (Chapter 6) is a selection of my publications directly related to the content of the first part. Most of the selected publications are joint work with other researchers, which is typical in the field. It goes without saying that I selected only papers where my contribution was essential.

1. Low-resource Languages

1.1 Dependency Parsing

The task of predicting the dependency tree structure for a previously unseen sentence is called **dependency parsing**. Nowadays it typically includes predicting the label (type) for each dependency relation, that is, for each word the **parser** must identify the word that should serve as its parent node, and the type of the relation between the two words. The parser is usually a model trained on manually annotated (**‘gold standard’**) data. The performance of a parser is evaluated on test (evaluation) data, which is separate from training data. The parser is applied to unannotated (**‘blind’**) version of the test data, and the parser’s output is then compared to manually annotated version of the same data. The most widely used evaluation method is the Labeled Attachment Score (**LAS**) – we count a word as correct if both its parent and the dependency type have been predicted correctly, and we compute LAS as the percentage of correct words among all words¹ in the test data. In situations where prediction of the labels is considered uninteresting or too difficult, Unlabeled Attachment Score (**UAS**) is used instead. It counts a word as correct if its parent has been identified correctly, ignoring the dependency label.

In my PhD thesis [Zeman, 2004], I explored dependency parsing of Czech. My parser² was not only result of several years of my own work; it also rested on the shoulders of a large team of colleagues who had spent over five years designing annotation rules and annotating 70 thousand Czech sentences on multiple levels. It struck me that the Czech language was very lucky to have such rich computational resources, far exceeding most languages of the world (including languages with far more speakers). Regardless that I tried to keep my parsing algorithm as language-agnostic as possible, I could not apply it to most languages simply because there was no training data. The situation has improved since then, but the problem of **low-resource languages** has not disappeared and it is not going to disappear any soon. There are thousands of natural languages in the world [Dixon, 2010, p. xiii] and if we now have about 100 languages with decent treebanks, there are still thousands of languages that lack them. I became interested in language processing that could be applied to many languages, including those that possess little or no hand-annotated data. I started to explore techniques of parsing a low-resource language B , taking advantage of better-resourced, related language A . For instance, could we build a reasonably performing parser for Slovak, given that it is very close to Czech, and while Slovak did not have any treebank, there was so much data available for Czech?

¹Most implementations of LAS work with all nodes, i.e., not only actual words, but also punctuation symbols and other tokens.

²With $UAS = 74.7\%$ on the d-test data of PDT 1.0 I fell significantly behind the state of the art (84.3%), but in combination with other parsers, my parser contributed to the new SotA $UAS = 85.5\%$.

1.2 Delexicalized Parsing

The technique I developed³ [Zeman and Resnik, 2008] (Section 6.1) was based on four simple assumptions:

- It is easier and thus cheaper to obtain gold-standard data with morphological tags than with syntactic structures.
- Languages that are related are likely to have similar syntactic structures, even if their lexical forms differ.
- A model can predict the syntactic structure reasonably well with only morphological tags (but not the actual word forms) as input.
- The sets of morphological tags for the related languages are mutually compatible.

We did not attempt to quantitatively evaluate the first assumption but it seemed quite intuitive, and it was supported by the existence of tagged corpora for many languages for which no treebank was available.

As for the second assumption, there are varying levels of relatedness. An obvious candidate is the genealogic relationship, with Czech being most closely related to other West Slavic languages (Slovak, Upper Sorbian and Polish), then to other Slavic languages, then to Baltic languages, then to other Indo-European languages. Languages can be *typologically* related because of common ancestry, but also because of geographic proximity and mutual interaction; for example, Bulgarian and Macedonian are in some aspects closer to Greek or Romanian than to other Slavic languages. But even distant languages may share some common traits, such as nouns being typical subject dependents of verbs.

To illustrate this, consider the sentence *My daughter tasted strawberry ice cream yesterday* in four Slavic languages (Figure 1.1). The Czech and Slovak versions are very close, even with half of the words identical. Ukrainian uses different words (and script) but the syntactic structure, as well as the sequence of part-of-speech tags is still the same. Polish slightly diverges from the other three languages in preferring the post-nominal position of the adjectival attribute; with that exception, its surface order mimics the other languages, and its dependency tree is still isomorphic with theirs.

A parsing model that relies on word forms can hardly be trained on one language and successfully applied to another – even the 50% of unknown words in Slovak could be devastating.⁴ However, if the parser can obtain most of the required information from part-of-speech tags, its Czech model will work just as well for the Slovak and Ukrainian sentence, and probably almost as well for Polish (we cannot rule out that it will predict the dependency of the ‘misplaced’

³This research was done during my stay at the University of Maryland in 2006. I am grateful for the interesting interactions with the colleagues there, in particular with Philip Resnik, under whose mentoring I did the work. I also acknowledge the funding provided jointly by the Fulbright-Masaryk Fellowship and by the Office of Naval Research.

⁴This motivational example should not be taken as a proof of anything. We have not provided evidence that the out-of-vocabulary rate will stay 50% on a larger data sample; we are just suggesting that the rate is not negligible.

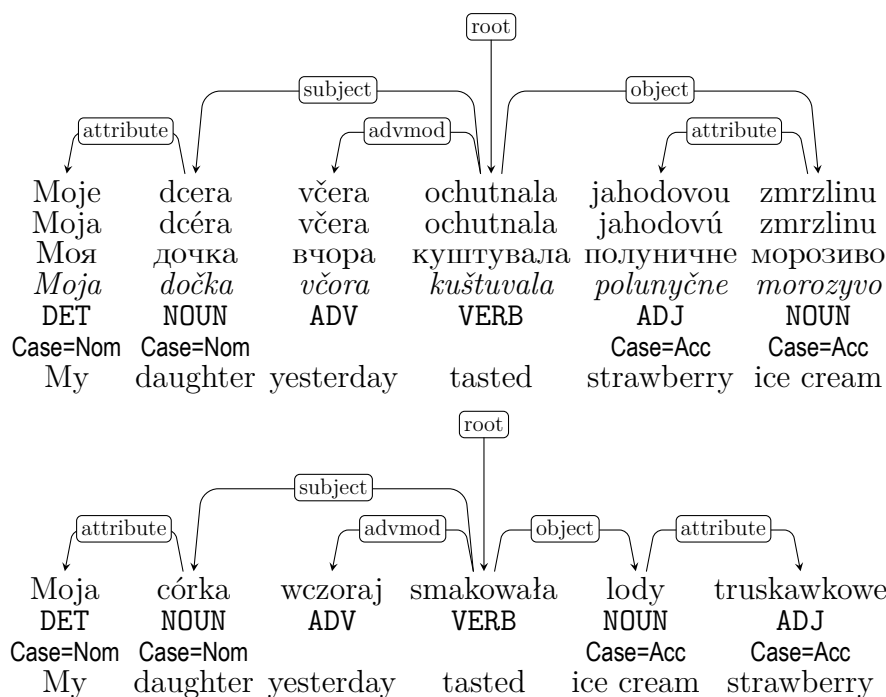


Figure 1.1: The sentence “*My daughter tasted strawberry ice cream yesterday*” in Czech, Slovak and Ukrainian (upper tree) and in Polish (lower tree).

adjective correctly). Even better if the tags are morphological, that is, if they reveal not only the part of speech but also the nominative case of words 1 and 2, and the accusative case of words 5 and 6.

The same part-of-speech sequence corresponds to many other sentences (or their parts) that have the same syntactic structure, for example

- [cs] *Tento sortiment také tvoří hlavní část [produkce společnosti]* “This assortment also forms the main part [of the company’s production]”
- [cs] *[...] jejíž část zatím nemá hlasovací právo [...]* “[...] part of which does not yet have voting rights [...]”
- [en] *All offices also have free copies*
- [pt] *A direção já mostrou boa vontade* “Management has already shown good will”
- [zh] 任何議員未曾作最後宣誓 (*Rèn hé yì yuán wèi céng zuò zuì hòu xuān shì*) “No member has taken his final oath”

This leads us to the third assumption, namely that morphological information is a sufficient characterization of the input words for a parser. Of course, there may be other sentences with the same sequence of tags whose syntactic structure is different. It is also clear that there are cases that cannot be decided without understanding the lexical content, as in the Czech examples below, where *v Ústí* is a modifier of the university, while *v září* modifies the event, i.e., the verb.

- [cs] *Přestoupím na univerzitu v Ústí* “I will move to the university in Ústí”
- [cs] *Přestoupím na univerzitu v září* “I will move to the university in September”

The best way of testing the seriousness of this deficiency is to train a parser and evaluate it using the standard attachment score (see Section 1.4). We call a model that has been trained only on morphological tags, without any lexical information, a **delexicalized parser**.

Finally, there is the fourth assumption, which may not be obvious from the start, nevertheless it is very important: We need the tag sets for the languages in question to be compatible, that is, the same part of speech or morphological feature should be encoded the same way in every language. As demonstrated in Table 1, this is rarely the case; in fact, even within one language different corpora may use different tag sets. I will address this issue in Chapter 2.

Delexicalized parsing was later explored by many other authors. Most notably, McDonald et al. [2011] conducted large-scale experiments with delexicalized parser transfer among 9 Indo-European languages, and they also combined delexicalized parsing with part-of-speech tag projection across parallel data (see Section 1.3), removing the requirement that a tagged corpus be available in the target language. Aufrant et al. [2016] improved delexicalized parsing by adapting word order before training the model (cf. the word order difference between Polish and the other three languages in Figure 1.1).

More recently [Kondratyuk and Straka, 2019], parsers started using large multilingual neural language models to represent the words and their context. These models can also consider subwords (even individual characters), which allows them, e.g., to assess that the Czech adjective *jahodovou* and the Slovak *jahodová* “strawberry” are equivalents. Such parsers can be viewed as occupying the middle ground between lexicalized and delexicalized. They have access to full lexical information, but they are also able to use it for an unknown word in a low-resource language if similarities can be observed on unannotated raw data.

1.3 Using Parallel Data

Other techniques that have been proposed for low-resource languages take advantage of **parallel texts**, that is, translations of the same text into multiple languages. They do not require that the text is annotated (specifically, morphological tags are not required). Again, the motivation is that unlabeled parallel texts are often available for pairs of languages where one language has rich annotated linguistic resources and the other does not. Indeed, there are many sources of such texts, ranging from multilingual legal documents (e.g., proceedings of the European Parliament) to open movie subtitles, to translations of the Bible.

Once a parallel corpus is available, unsupervised algorithms, well known from the machine translation field, can be used to first align sentences that are translations of each other, and then for each pair of parallel sentences compute the word alignment. The alignments provide links between elements of sentence structure, and these links can be used to project linguistic annotation from the resource-rich to the resource-poor language. As with delexicalized parsing, the techniques can

be applied to any pair of languages, but better results are expected for languages that are closely related.

Training data projection. Run the source-language model on the source side of the parallel data, annotate it automatically. Project the annotation across word alignments to the target side of the parallel data. Train a target-language model on the now annotated target side of the parallel data.

Training data translation. Use the parallel data to obtain a simple word-to-word translation model. Apply it to the source-language annotated data. As a result, we have a ‘translated’ target-language corpus with exactly the same number of words, hence we can directly use the source-language annotation with the target-language word forms. Train a target-language model on the translated data. Of course, this technique makes sense only for closely related languages.

Test data translation. Use the parallel data to obtain a simple word-to-word translation model. Apply it to the target-language blind test data. Once ‘translated’ to the source language, we can apply the source-language model to annotate the data. Then the text can be ‘re-stuffed’ with the original target words, and used for whatever purpose we needed the annotation. This resembles delexicalized parsing but instead of replacing the words with morphological tags, we replace the words with their equivalents in the other language.

Training data projection for part-of-speech tagging was first proposed by Yarowsky and Ngai [2001] and later refined by other authors. Das and Petrov [2011] used a word lattice in the target language to propagate tags to words that did not occur in the parallel data but were similar to words from the parallel data in that they preferred similar context. Agić et al. [2015] showed that part-of-speech projection is available for a large number of languages thanks to translations of the Bible. Mishra et al. [2017] experimented with “feature projection” for part-of-speech tagging of Indian languages. Their technique is similar to word-by-word translation of the training data.

Concerning dependency parsing, training data projection was proposed by Hwa et al. [2005]. In [Zeman and Resnik, 2008], we experimented with test data translation for dependency parsing and compared it to delexicalized parsing. The results we obtained spoke in favor of delexicalized parsing, but the translation approach fell not too far behind and it should not be ruled out for other datasets. Tiedemann [2014], Ramasamy [2014], Rosa [2018] compared the advantages and disadvantages of the projection and translation techniques. In 2017 our team won the shared task on similar language parsing [Rosa et al., 2017];⁵ we used a variant of training data translation.

Annotation projection across parallel data has been applied even beyond surface syntax, for example to semantic roles that were projected from the English PropBank to several other languages [Jindal et al., 2022].

⁵The task consisted of parsing three target languages: Slovak (with Czech as the source language), Croatian (with Slovenian as the source), and Norwegian (with two source languages, Danish and Swedish). This shared task provided harmonized annotations for the languages in question.

1.4 Evaluation

The cross-lingual techniques outlined in the previous sections are useful if we do not have manually annotated data in the target language. However, in order to evaluate the performance of the techniques, we do need target gold-standard data. The evaluation is thus typically conducted on languages that possess annotated corpora, using those corpora only for evaluation, and hoping that the method would work similarly well when applied to a really resource-poor language. Once again, we need the annotation in the source and target languages to be compatible. If we are projecting parsing models, the compatibility requirement applies also to dependency trees – the rules for positing a dependency relation between two words, and the label (type) of the relation. None of that is granted (recall Figure 2); in fact, the opposite was the norm until about 2012.

The first CoNLL shared task in multilingual dependency parsing [Buchholz and Marsi, 2006] made available dependency treebanks of 13 languages.⁶ The datasets were unified technically, using the same file format (later dubbed CoNLL-X), but their label sets were not harmonized, and neither were the linguistic decisions governing the dependency relations. On the other hand, the collection provided an opportunity to test cross-lingual transfer of parsers, as it included two closely related languages: Danish and Swedish.

The Danish data followed the annotation guidelines of the Danish Dependency Treebank [Kromann, 2002], while the Swedish data was taken from Talbanken [Nilsson et al., 2005]. These two treebanking schemes are very distant from each other. In [Zeman and Resnik, 2008], we employed various heuristics to make the annotations comparable; then we used Danish as the source language and Swedish as the target language. In contrast, McDonald et al. [2011] did not attempt to harmonize their data, and their results picture Danish as the worst possible source language for Swedish, among the eight European languages available.⁷

The actual attachment scores can be found in the respective papers cited here. They are not directly comparable, as they have been obtained on diverse datasets of various languages, and also with many different parsers (note that the delexicalization, projection and translation techniques can be used with any parser that can be trained on annotated data). Roughly speaking, one can expect around 65% UAS for closely related languages, meaning that two out of three words have the correct parent node. An interesting perspective to view this number is provided by a comparison with the learning curve of a fully supervised parser. The question we ask is: If manual annotations were available for the target language, how much of them would we need to train a parser that performs as well as our model transferred from the source language? Hwa et al. [2005] showed that their projection from English to Chinese corresponded to about 2000 Chinese gold-standard trees. The best Danish-based model from [Zeman and Resnik, 2008] ranked equal to a parser trained on 1546 Swedish sentences. I repeated the experiment in 2015 with more advanced parsers and better harmonized data. The UAS was still 66% but the learning curve was steeper, suggesting that the

⁶Not all the treebanks were available free of charge after the shared task.

⁷There were four other Germanic languages in the mix but none of them worked well, presumably also due to annotation divergences. The most helpful source, as evaluated on the Swedish data, turned out to be the Portuguese treebank.

same result can be obtained with just 75 Swedish sentences. Along the same lines, Ramasamy [2014, Table 6.6 on p. 100] found that with just 10 annotated training sentences, the UAS on his language set ranges from 57% (Bengali and Tamil) to 74% (Telugu) on in-domain target language data. Therefore, if a native speaker of the target language is available for a few days, the best technique might be to have the native speaker annotate a small sample of the target language. But this approach does not scale well to hundreds or thousands of target languages.

At any rate, we need annotations to be **harmonized** across languages in order to train and evaluate multilingual NLP tools, regardless of what particular approach we take. We will focus on harmonization in the following chapters.

2. Harmonization of Morphological Annotation

2.1 Interset

In Chapter 1, I stressed the necessity of working with corpora that have mutually compatible annotation. Specifically, for delexicalized parsing I needed a morphological tagset that could be applied to both the source and the target language. Since each of the available corpora used its own tagset, I had to either convert tags from tagset A to tagset B , or to define a hybrid tagset C covering features that are common to both corpora, and then convert A and B to C . While we described experiments with Danish and Swedish in [Zeman and Resnik, 2008], I conducted similar experiments with other language pairs, which means many different conversions had to be done. A typical conversion procedure is based on a large table or on a long sequence of `if-else` statements, and preparing it is tedious work. Therefore I was looking for ways how to reuse parts of the code written previously. Each conversion from tagset A to tagset B can be viewed as two steps done at once: understanding the information in tag A (**decoding**) and producing tag B that contains same or similar information (**encoding**). If I separate the steps, I will be able to reuse them in the future when I encounter a new tagset C and need conversion from A to C , or from C to B . I will only have to implement the decoder and encoder for tagset C ; then I can immediately convert tags between C and any previously covered tagset. I implemented this mechanism in Perl, and the Perl modules with encoders and decoders for individual tagsets were called **tagset drivers** [Zeman, 2008, 2018] (Section 6.2).

A crucial part of the conversion system is the intermediate feature structure where the information is stored between decoding from tagset A and encoding to tagset B . It functions as an Interlingua for morphological tagsets and I named it **Interset**.¹ Information from a morphological tag was decomposed and stored as a set of pre-defined morphological features (such as `pos` (part of speech), `gender`, `number`, `tense`) and one of their pre-defined values (such as `pos=noun` or `tense=past`). Interset turned out to be a useful framework for describing morphosyntax independently of individual corpora; as such, its significance grew beyond the engineering problem of preparing data for an experiment.

Conversion of a tag to a different tagset is often an information-losing process because the tag may make distinctions that the target tagset does not make. Nevertheless, we do not want to lose information during round-trip ‘conversion’ from a tagset to itself (i.e., when Interset is used as an internal data structure to easily access information about words, without the need to actually convert the tag). It may not be possible to capture all distinctions in a tagset because some of them may be too peculiar to deserve an Interset feature. Therefore, a decoder can always store additional data to a feature called `other`. The data is not expected to be understood by any other driver, hence Interset also remembers the identifier of the source tagset in the feature `tagset`. The encoder will consult

¹By extension, ‘Interset’ also refers to the conversion software built around the data structure (<https://ufal.mff.cuni.cz/interset>).

the value of `other` only if it originates in the same tagset.

Interaset was built bottom-up and new features or values were occasionally added when they were needed for newly added tagsets. If the existing feature-value pairs could not capture something in a new tagset, I had to assess whether it was worth adding a new feature (or value). If not, then it would be stored in `other`. In some cases, a feature was first stored in `other` but later revisited and made a regular Interaset feature, when it was attested in another tagset.

In the current version, Interaset covers 64 tagsets of 40 languages. It defines 63 features with 390 values in total. Some of the features are lexical, that is they pertain to the whole lexeme with all its morphological forms; they can be viewed as a finer partition of the part-of-speech space. Other features are inflectional, they describe the position of an inflected word form in the lexeme’s inflectional paradigm. This classification is only approximate, for example, `gender` is lexical feature of Czech nouns but inflectional feature of Czech adjectives. However, the lexical-inflectional distinction serves only for orientation purposes and has no practical impact on work with Interaset. Similarly, one could classify features as typically nominal (e.g., `case`) or typically verbal (e.g., `tense`), but many features would combine with multiple parts of speech, and plausible combinations would vary across languages (for example, Czech verbs do not inflect for `case` but some forms of Finnish verbs do).

Table 2.1 gives an overview of features and values in the current version of Interaset together with a brief explanation of each feature.

Table 2.1: Interaset features and their values.

<code>pos</code>	noun, adj, num, verb, adv, adp, conj, part, int, punc, sym	main part of speech
<code>nountype</code>	com, prop, class	special type of noun if applicable
<code>nametype</code>	geo, prs, giv, sur, nat, com, pro, oth, col, sci, che, med, tec, cel, gov, jus, fin, env, cul, spo, hob	named entity type
<code>adjtype</code>	pd	special type of adjective: pre-determiner
<code>prontype</code>	prn, prs, rcp, art, int, rel, exc, dem, emp, neg, ind, tot	pronominality and its type for nouns (pronouns), adjectives (determiners), numerals, adverbs
<code>numtype</code>	card, ord, mult, frac, sets, dist, range	numeral types; the main pos may be numeral, adjective, adverb
<code>numform</code>	word, digit, roman, combi	presentation form of numerals
<code>numvalue</code>	1, 2, 3	class of numeric values for numerals with special behavior
<code>verbtype</code>	aux, cop, mod, light, verbconj	special type of verb if applicable
<code>advtype</code>	man, loc, tim, sta, deg, cau, mod, adadj, ex	semantic type of adverb
<code>adpostype</code>	prep, post, circ, voc, prepron, comprep	special type of adposition if applicable
<code>conjtype</code>	coor, sub, comp, oper	conjunction type
<code>partype</code>	mod, emp, res, inf, vbp	particle type

Continuation of Table 2.1

punctype	peri, qest, excl, quot, brck, comm, colo, semi, dash, root	punctuation type
puncside	ini, fin	distinction between opening and closing brackets and other paired punctuation
morphpos	noun, adj, pron, num, adv, mix, def	morphological part of speech – inflectional paradigm may behave like different pos than the word is assigned to
poss	yes	possessive word
reflex	yes	reflexive word
foreign	yes	foreign word
abbr	yes	abbreviation
hyph	yes	part of a hyphenated compound
typo	yes	incorrect form
echo	rdp, ech	reduplicated or echo word
polarity	pos, neg	polarity: affirmative or negative
definite	ind, spec, def, cons, com	definiteness and/or construct state
gender	masc, fem, com, neut	gender
animacy	anim, hum, nhum, inan	animacy
number	sing, dual, tri, pauc, grpa, plur, grpl, inv, ptan, coll, count	grammatical number
case	nom, gen, dat, acc, voc, loc, ins, abl, del, par, dis, ess, tra, com, abe, ine, ela, ill, ade, all, sub, sup, lat, per, add, tem, ter, abs, erg, cau, ben, cns, equ, cmp	grammatical case
prepcase	npr, pre	special case form after a preposition
degree	pos, cmp, sup, abs, equ, dim, aug	degree of comparison; also diminutives and augmentatives
person	0, 1, 2, 3, 4	person
clusivity	in, ex	inclusive vs. exclusive pronoun <i>we</i>
polite	infm, form, elev, humb	politeness, formal vs. informal word forms
possgender	masc, fem, com, neut	possessor's gender
possperson	1, 2, 3	possessor's person
possnumber	sing, dual, plur	possessor's number
possednumber	sing, dual, plur	possession's number; in Hungarian distinguished from main number and possessor's number
absperson	1, 2, 3	person of the absolutive argument of the verb (polypersonal agreement in Basque)
ergperson	1, 2, 3	person of the ergative argument of the verb (polypersonal agreement in Basque)

Continuation of Table 2.1

datperson	1, 2, 3	person of the dative argument of the verb (polypersonal agreement in Basque)
absnumber	sing, dual, plur	number of the absolutive argument of the verb (polypersonal agreement in Basque)
ergnumber	sing, dual, plur	number of the ergative argument of the verb (polypersonal agreement in Basque)
datnumber	sing, dual, plur	number of the dative argument of the verb (polypersonal agreement in Basque)
abspolite	infm, form, elev, humb	politeness of the absolutive argument of the verb (polypersonal agreement in Basque)
ergpolite	infm, form, elev, humb	politeness of the ergative argument of the verb (polypersonal agreement in Basque)
datpolite	infm, form, elev, humb	politeness of the dative argument of the verb (polypersonal agreement in Basque)
erggender	masc, fem, com, neut	gender of the ergative argument of the verb (polypersonal agreement in Basque)
datgender	masc, fem, com, neut	gender of the dative argument of the verb (polypersonal agreement in Basque)
position	prenom, postnom, nom, free	position / usage of adjectives, determiners, participles etc.
subcat	intr, tran	subcategorization (transitive vs. intransitive)
verbform	fin, inf, sup, part, conv, vnoun, ger, gdv	finite verb vs. infinitive, supine, participle, converb, verbal noun, gerund, gerundive
mood	ind, imp, cnd, pot, sub, jus, prp, opt, des, nec, qot, adm	mood
tense	pres, fut, past, aor, imp, pqp	tense
voice	act, mid, pass, rcp, cau, int, antip, dir, inv	voice
evident	fh, nfh	evidentiality
aspect	imp, perf, prosp, prog, hab, iter	aspect (lexical or grammatical)
strength	weak, strong	strong vs. weak forms of adjectives or pronouns
variant	short, long, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c	variant form of the same lemma and paradigm slot

Continuation of Table 2.1

style	arch, rare, form, poet, norm, coll, vrnc, slng, expr, derg, vulg	style (either of the lemma, or standard vs. colloquial suffix of the same lemma)
tagset	e.g. cs:pd	source tagset identifier (determines relevance of other)
other	any value, possibly structured	tagset-specific information that does not fit elsewhere

2.2 ‘Google’ Universal POS Tags

A few years after the first version of Intersect, a team from Google and Carnegie-Mellon University proposed a set of 12 universally applicable and universally needed, coarse-grained part-of-speech tags for use in NLP applications [Petrov et al., 2012]; this tagset became informally known as the ‘Google’ universal tagset. Their goal was to harmonize the encoding of the main categories of words, ignoring finer morphological distinctions. In Intersect, they would approximately correspond to the eleven non-empty values of the `pos` feature.

The authors also offered mappings from 25 existing tagsets of 22 languages to the universal tagset. An important shortcoming of their approach in comparison to Intersect was that their mappings often relied exclusively on the top-level part of the source tag. So, for example, they defined a tag for numerals (NUM), but the source tagset for Danish did not have numerals as a top-level category. Instead, they were treated as a subclass of adjectives and consequently, they would end up as ADJ in the universal tagset, although by looking at other parts of the Danish tag, one could actually tell apart numerals from adjectives. Some of these issues were fixed in later versions of the mapping tables.²

2.3 Universal Dependencies

Having one annotation standard that fits all languages and applications is obviously beneficial for natural language processing. Also obviously, having more than one standard reduces the benefit. On the morphological level, there were universal POS tags, Intersect, and some older standardization attempts which I survey in [Zeman, 2008]. There were at least two harmonization efforts also on the syntactic level (more on that in Chapter 3). In 2014, we joined forces with colleagues from Uppsala University, Stanford University, Google, University of Turku, Bar-Ilan University and the Open University of Israel. Our goal was to take the best from the previous harmonization efforts and try to build one standard that would supersede them. The team included authors of the competing harmonization projects, which was one important ingredient for success. The name of the new framework, Universal Dependencies³ [de Marneffe et al., 2021] (Section 6.4), refers to syntactic annotation, but the framework defines cross-linguistic annotation both for syntax and morphology.

²<https://github.com/slavpetrov/universal-pos-tags>

³<https://universaldependencies.org/>

Universal Dependencies (UD) uses an extended version of the Universal POS tagset, now also abbreviated UPOS, with 17 tags instead of the original 12 (the additions included **PROPN** for proper nouns, **AUX** for auxiliaries, **SCONJ** for subordinating conjunctions, **INTJ** for interjections, and **SYM** for symbols other than punctuation. Besides UPOS, the UD standard has morphological features. The core set of features and values, documented as “universal features”, are taken from Interset.⁴ UD corpora can extend that set with their own features if needed, and some of the remaining Interset features have been used this way. I continue to maintain the feature set within the UD project and occasionally propose language-specific features or values, when they are attested in multiple corpora, to be promoted to the universal features. This ensures that people working on new languages for UD will use those features if they apply to their language, following the objective that same things be annotated same way in all languages. Interset proper still exists as a tagset conversion tool and I keep it compatible with UD.

2.3.1 Layered Features⁵

In some languages, some features are marked more than once on the same word. For example, possessive pronouns (also called possessive determiners or adjectives in various terminological systems) may have two independent values of gender and two independent values of number. One of the values characterizes the possessor, the other characterizes the possessee. The possessor’s gender and number is something that we observe also with normal personal pronouns: for instance, the English 3rd-person pronouns distinguish singular and plural, and they also distinguish three genders in the singular (*he, she, it*) but not in the plural (*they*). Likewise, the corresponding possessive pronouns have three genders in singular (*his, her, its*) but only one form in plural (*their*). English does not mark the possessee’s features morphologically, but other languages do.

Thus in Croatian, the 3rd person pronouns distinguish three genders and two numbers in the nominative case, but in the other cases and in the possessives, the singular masculine is often identical to the singular neuter, and the plural forms are mostly common for all three genders. In most cases, there are three distinct forms (Table 2.2). There are also possessive pronouns for three different categories of possessors: masculine/neuter singular (*njegov*), feminine singular (*njezin*),⁶ and plural (*njihov*). However, in Croatian the possessive pronouns behave like adjectives and agree in gender, number and case with the possessed (modified) noun. If the possessee is masculine singular, such as *pas* “dog”, the possessive pronoun will acquire a masculine suffix: *njegov pas* “his dog”, *njezin pas* “her dog”, *njihov pas* “their dog”. If the possessee is feminine singular, the form of the possessive changes and takes the feminine suffix: *njegova mačka* “his cat”, *njezina mačka* “her cat”, *njihova mačka* “their cat”. Similarly for singular neuter (*njegovo polje* “his field”), plural masculine (*njegovi psi* “his dogs”) etc.

We thus need tags that distinguish the ordinary agreement suffixes (i.e., the possessee’s gender, number and case) from the possessor’s gender and number,

⁴Only capitalization is changed, e.g. the Interset feature **gender=masc** is **Gender=Masc** in UD.

⁵Subsection first published in Zeman [2018], reproduced here only with minor changes.

⁶In fact, there is a second feminine possessive variant: *njen*. We disregard it here.

Case		Sing Masc/Neut	Sing Fem	Plur Masc/Fem/Neut
Prs	Nom	<i>on/ono</i>	<i>ona</i>	<i>oni/one/ona</i>
Prs	Gen	<i>njega</i>	<i>nje</i>	<i>njih</i>
Number Gender Case				
Poss	Sing Masc Nom	<i>njegov</i>	<i>njezin</i>	<i>njihov</i>
Poss	Sing Fem Nom	<i>njegova</i>	<i>njezina</i>	<i>njihova</i>
Poss	Sing Neut Nom	<i>njegovo</i>	<i>njezino</i>	<i>njihovo</i>
Poss	Plur Masc Nom	<i>njegovi</i>	<i>njezini</i>	<i>njihovi</i>
Poss	Plur Fem Nom	<i>njegove</i>	<i>njezine</i>	<i>njihove</i>
Poss	Plur Neut Nom	<i>njegova</i>	<i>njezina</i>	<i>njihova</i>

Table 2.2: The nominative and genitive forms of Croatian 3rd person pronouns, and the nominative forms of the corresponding possessive pronouns. The rows represent various genders and numbers of the possessee, while the columns represent genders and numbers of the possessor.

which is encoded in the stem. Universal Dependencies call this *layered features*: there are two layers of gender, and two layers of number. There is also a specific notation: if a word is annotated more than once with a feature, the layers must be identified by a predefined string given in square brackets. For instance, a masculine possessor would be annotated as `Gender[psor]=Masc`. One layer can be treated as default and given without layer name; in our example, the agreement gender would be annotated simply as `Gender=Masc`. Note that Intersect did not have such a flexible mechanism and had to define a separate feature for each layer. For instance, UD’s `Gender[psor]` corresponds to `possgender` in Table 2.1. Another example where layered features help is polypersonal agreement in languages like Basque: when morphology of a ditransitive verb concurrently refers to three arguments distinguished by the absolutive, ergative and dative case, Intersect would encode the verbal agreement as `absperson`, `ergperson` and `datperson`, while the layers in UD would lead to `Person[abs]`, `Person[erg]` and `Person[dat]`.

2.4 UniMorph

For completeness I also briefly mention another project that tries to capture morphology across languages: UniMorph. It started independently of UD, shortly after the first version of UD was released [Sylak-Glassman et al., 2015]. It took a top-down approach, trying to survey the known morphological categories from typological literature and project them all to the schema even before they were actually seen in corpora. Fortunately, UniMorph did not lead to a new competition between standards of morphological annotation. I took the proposal into account when designing the second version of the UD guidelines in 2016 and adopted some features that had been defined in UniMorph but not in UD. The two frameworks use similar level of granularity, and although they do not align perfectly, most UniMorph features can be represented in UD without loss of information. UniMorph and UD are now overlapping communities that take care to minimize potential incompatibilities between the two schemas.

3. Harmonization of Syntactic Annotation

3.1 HamleDT

I showed some examples of diverging approaches to syntactic annotation in Figures 1 and 2 in the Introduction, and in Section 1.4, I reported on experiments where the benefits of close relationship between Danish and Swedish were negated by the differences in the annotation of Danish and Swedish data. In [Zeman and Resnik, 2008] I used simple transformation heuristics to make the Danish and Swedish treebanks more comparable. However, this was an ad-hoc solution that did not consider datasets of other languages and did not lead to harmonized annotations that other researchers could reuse. In 2011, I and several my colleagues from Charles University decided to find a more principled and far-reaching solution.

We first inventoried the various dependency treebanks that were available at that time, and studied their annotation styles. To demonstrate the differences, in Figures 3.1–3.6 I show the coordination structure *apples, oranges and lemons* annotated according to 6 different treebanking styles.¹

We implemented a technical conversion to a common file format – we used the CoNLL-X format defined by Buchholz and Marsi [2006], which had already become a de-facto standard used by various NLP tools. The morphological tags were converted to Intersect features and stored in the file. Then we implemented transformations of the dependency structures.

It was almost a rule that each treebank had its own annotation style. An exception to this rule was a group of about ten treebanks inspired by the Prague Dependency Treebank [Hajič et al., 2000]; their annotation styles were not identical but they were reasonably similar. Since PDT was the home product of our institute, we naturally based our common annotation scheme on PDT. We named the collection HamleDT² (Harmonized Multi-Language Dependency Treebank) [Zeman et al., 2014] (Section 6.3). Its first version [Zeman et al., 2012] covered 29 languages but we later expanded it to 36 languages.

3.2 Stanford Dependencies

Another dataset with common annotation scheme was made available by a team of researchers from Google and Appen [McDonald et al., 2013]. Its first version contained six languages: English and Swedish were conversions of datasets that we also had in HamleDT; Spanish, French and Korean were newly annotated texts collected from the web, and German combined a pre-existing treebank with new data from the web. A year later the collection was expanded to 11 languages.³ The authors called it ‘Universal Dependency Treebank’; to distinguish it from

¹See Popel et al. [2013] for more details on coordination styles in treebanks.

²<https://ufal.mff.cuni.cz/hamledt>

³<https://github.com/ryanmcd/uni-dep-tb>

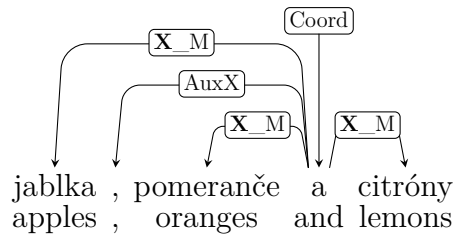


Figure 3.1: Coordination in the Prague style as seen in the Prague Dependency Treebank of Czech. **X** represents the relation between the coordination and its parent in the sentence.

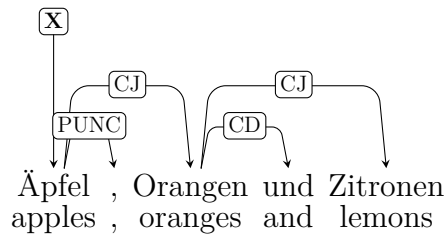


Figure 3.2: Coordination in the Mel'čukian style as seen in the Tiger treebank of German.

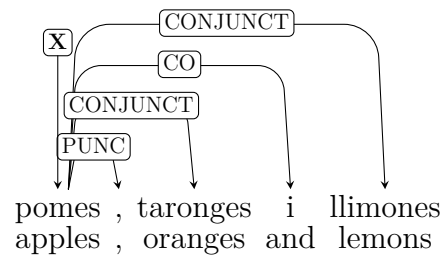


Figure 3.3: Coordination in the Stanford style as seen in the AnCora treebank of Catalan.

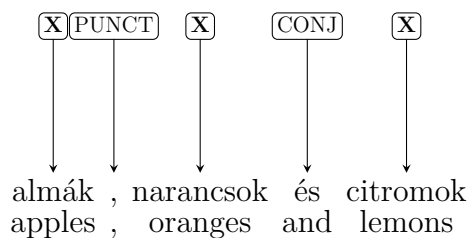


Figure 3.4: Coordination in the Tesnièrean style as seen in the Szeged Treebank of Hungarian. All participating nodes are attached directly to the parent of the coordination.

the Universal Dependencies project, it is sometimes informally dubbed ‘Google’ Universal Dependency Treebank. At the morphological level, it used the Google universal POS tags without additional features. At the syntactic level, they used a variant of Stanford Dependencies (SD) [de Marneffe et al., 2013]. As they said, the Stanford typed dependencies, partly inspired by the LFG framework, had emerged as a de-facto standard for dependency annotation in English and had

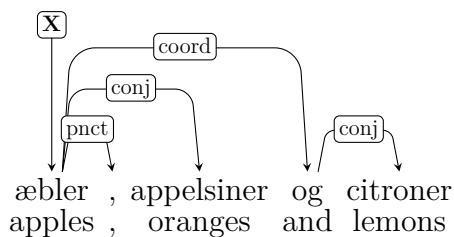


Figure 3.5: A mixture of Stanford and Mel’čukian coordination styles seen in the Danish Dependency Treebank.

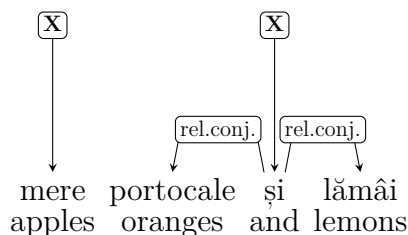


Figure 3.6: The Romanian treebank used Prague coordination style mixed with Tesnièrean because punctuation was missing from data.

then been adapted to several other languages; hence they decided to take SD as the point of departure for their representation.

The research group at Stanford University further developed their formalism to make it less biased towards English and more applicable to typologically diverse languages; the new proposal was called Universal Stanford Dependencies (USD) [de Marneffe et al., 2014]. In Prague, we noticed the growing popularity of Stanford-derived schemes and released HamleDT 2.0 with every treebank converted to two alternative schemes: Prague (based on PDT) and Stanford (based on USD) [Rosa et al., 2014].

3.3 Universal Dependencies

So in mid 2014 the problem of many diverging treebanks was replaced by the problem of several diverging standards, each of them hoping to solve the former problem. There were the Prague-style dependencies of HamleDT, and at least two flavors of the Stanford dependencies: the ‘Google’ flavor in the Google Universal Dependency Treebank, and the USD. In addition, there were Google UPOS and InterSet on the morphological level. As I already outlined in Section 2.3, our ultimate answer to this muddle was Universal Dependencies [de Marneffe et al., 2021] (Section 6.4). In the present section I will focus on the syntactic aspects of UD. Unlike morphology, the syntactic part of the UD standard was not derived from my previous work. Nevertheless, as a founding member of the UD core group I contributed to its development, in particular to the formulation of the second version of the standard in 2016 [Nivre et al., 2020].

The syntactic structures in UD are based on a modification of the Universal Stanford Dependencies. Both USD and UD try to maximize parallelism in annotation of the same construction across languages. This naturally leads to

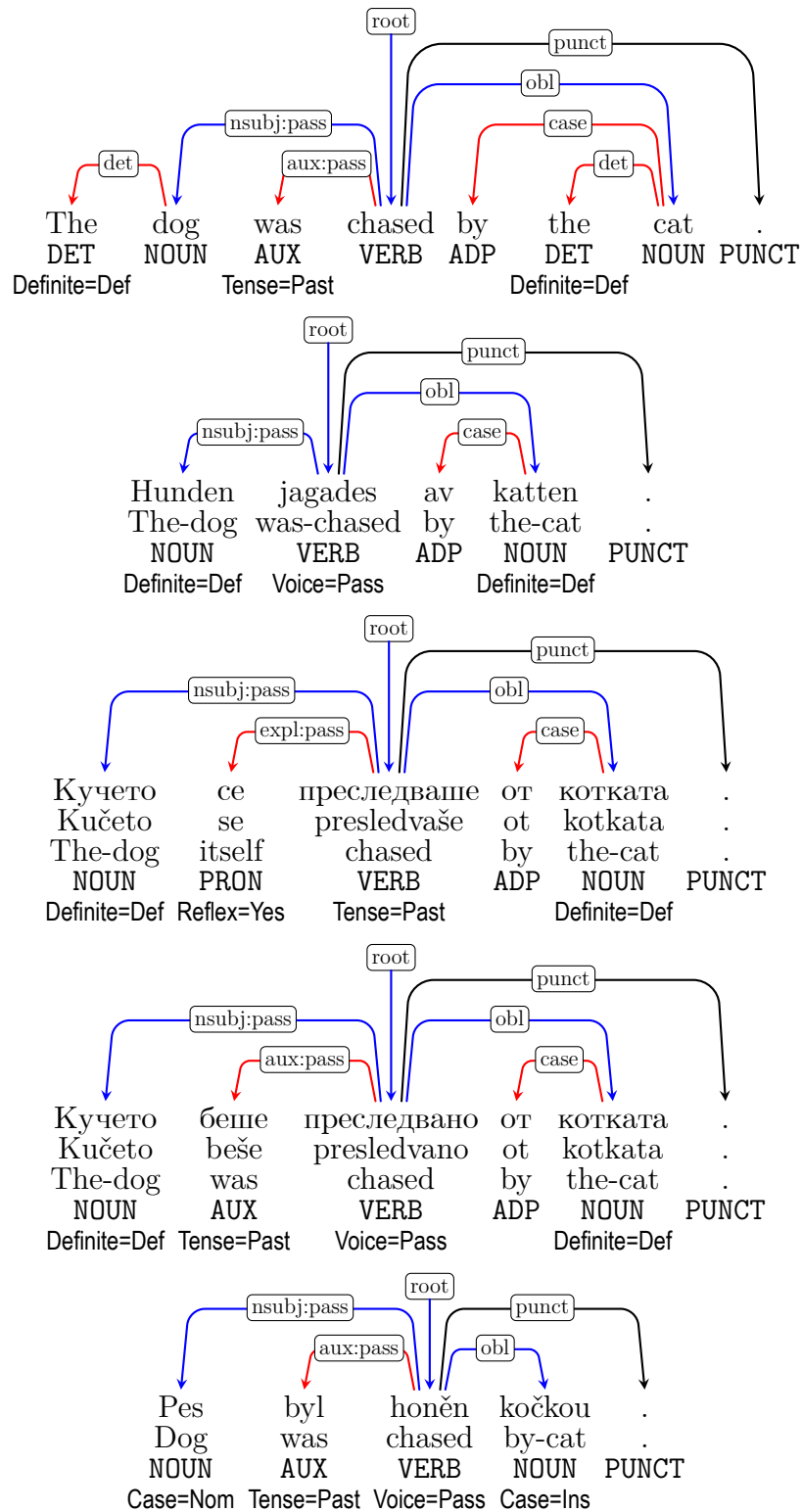


Figure 3.7: Parallel UD trees for the sentence *The dog was chased by the cat* in English, Swedish, Bulgarian (two versions) and Czech. Relations leading to content words are highlighted in blue, relations to function words in red and punctuation in black. Only selected features are shown.

preferring relations that place content words higher in the tree. Function words, which are more likely to vary across languages, are typically represented by leaf nodes. If we compare two languages where a function word in one language corresponds to a morphological feature in the other, the lexical backbones of the two trees stay parallel. This is demonstrated on the parallel sentences in Figure 3.7. The main meaning is expressed by the passive predicate *chase*, its subject *dog* and oblique agent *cat*; the relations between these three nodes are identical in all five trees. Relations attaching function words vary but they do not disrupt the main structure because their dependents are leaves. So in English there are separate nodes for the definite articles, while in Czech definiteness is not marked and in Swedish and Bulgarian it is marked directly on nouns. The oblique agent is marked by preposition in all languages but Czech, which uses the instrumental case (morphology). The passive voice is encoded with the help of an auxiliary in English, Czech and the second Bulgarian translation, by a reflexive pronoun in the first Bulgarian translation, and morphologically on the main verb in Swedish.

There were numerous typologically interesting constructions from many languages that we had to study when designing the UD guidelines. No doubt there are many others we will encounter as new languages and language families get covered by UD. I am not going to survey such constructions now because I have done so in [de Marneffe et al., 2021, § 4], which is incorporated in Section 6.4 of this thesis.

Universal Dependencies is a thriving project and community, which keeps growing and adding annotated resources for several new languages every year. In many cases UD literally helped to “put the language on the digital map.” UD treebanks are used in natural language processing but also in various areas of digital humanities, in particular linguistics and linguistic typology. While UD treebanks are probably too small to study the language system, parsers trained on these treebanks can be used to process additional data, often with decent accuracy. UD includes quite a few classical languages such as Ancient Greek or Sanskrit, thus aiding historical studies. Diversity of the collection is further increased by fieldworkers who create treebanks while documenting endangered languages (for example, we have samples of 15 indigenous languages from South America). The success of UD may lay in various factors which are difficult to evaluate, but the crucial point is that we tried to balance different perspectives and needs, however conflicting they may be. We tried to make it linguistically adequate but still simple enough for non-linguists, we built it on de-facto standards, kept the guidelines relatively stable over time, and maintained a regular cycle of two releases per year. This, together with the supporting infrastructure, makes it easy for newcomers to start a treebank and see it become part of UD in relatively short time. And once UD became known in the NLP community, the snowball effect went off: People who did not see their language in UD decided to do something about it and started annotating data. That is why we now⁴ cover 148 languages from 31 families and all parts of the world, the combined size of the treebanks exceeds 31 million words, it exists thanks to 577 contributors and it has cumulatively reached nearly 200 thousand downloads.

⁴UD release 2.13 from November 2023.

4. Multilingual Shared Tasks

It is a tradition in the field of natural language processing to organize evaluation campaigns – shared tasks – focused on concrete NLP problems. Such tasks serve multiple purposes. They help establish what is the current state of the art of solving the problem at hand on a given dataset; they typically also lead to advancing the state of the art by the best systems developed by task participants. In many cases, the evaluation data used in the task are also a new contribution, available to the research community after the task.

I have already mentioned (Section 1.4) the importance of the CoNLL 2006 and 2007 tasks for the area of multilingual dependency parsing. Now it is natural to ask how the parsing accuracy would change when parsers are evaluated on the annotation schema of Universal Dependencies. We thus decided to organize a new series of parsing shared tasks at CoNLL 2017 [Zeman et al., 2017] and 2018 [Zeman et al., 2018] (Section 6.5).

The algorithms of machine learning and dependency parsing had improved since 2007, so even a mere repetition of the 2007 task would have been interesting. However, our tasks were novel and brought new insights in a number of ways:

- Thanks to the uniform annotation scheme, it was now possible to compare parsing results across languages.
- It was now possible to combine training data from different languages to increase the robustness of parsing models. Participants were able to take advantage of data combination for well-resourced languages (e.g., a Swedish parser gave better results if it also saw Danish and Norwegian data besides Swedish), but it was especially useful for languages with little or no training data.
- To encourage multilingual and crosslingual parsing techniques, we included several low-resource languages, some of them without any training data. In the 2017 task we even introduced four ‘surprise languages’ (Buryat, Kurmanji, North Sámi, and Upper Sorbian) that had not been previously released in UD and the participants only got their names and a small data sample shortly before the test phase of the task. The default approach taken by the participants to such languages was a delexicalized parser (Section 1.2) trained on another language, but more successful were lexicalized models trained on multiple languages with weights for individual training datasets.
- An annotation effort was launched that yielded new parallel UD test sets (PUD), consisting of 1000 sentences from online news and Wikipedia, translated into 18 languages. Although this treebank collection was first used for parser evaluation in the shared task, it was later used in various contrastive studies, taking advantage of having the same contents with same annotation scheme in multiple languages.
- In addition to annotated treebanks, we also collected and made available large raw text corpora in 45 languages from Common Crawl to help the participants obtain word embeddings for their parsers.

- With a total of 82 test sets for 57 languages, the 2018 task became the largest and most multilingual evaluation campaign in dependency parsing to date. It set a new trend in NLP that tools and algorithms should be evaluated on large and typologically diverse sets of languages.
- Unlike the older parsing tasks, ours were designed as ‘end-to-end’ tasks, meaning that the submitted systems could not rely on gold-standard sentence segmentation, tokenization or part-of-speech tags in the input. We effectively redefined the standard setup of a parsing task. Before 2017 it would be common to assume that sentences and tokens are given;¹ since our shared tasks it is expected that a parser should be able to process raw text, which is more like a real-world scenario. Moreover, we also evaluated predicted POS tags and morphological features in the system output. These annotations, while interesting for human users, are typically not needed by modern parsers to predict the syntactic structure; by making them part of the evaluation we encouraged the participating systems to become full-fledged analyzers of natural language morphology and syntax.

With 32 participating teams in 2017 and 25 in 2018, the shared tasks can be considered a success. They also set the stage for a significant flow of follow-up research where multilingual parsing systems were evaluated using the same methodology and same type of data (the latest release of UD).

As cross-linguistic comparison of parsers was one of the goals of the shared tasks, we paid a lot of attention to comparability of the evaluation scores. The uniform annotation scheme was a necessary condition, but not a sufficient one. The standard labeled attachment score (LAS) is affected by various language-specific factors, such as the number of function words. The same grammatical meaning may be encoded by function words, by morphology, or not encoded at all; and while attachment of function words would be reflected in LAS, errors in morphological features would not. This is illustrated in Figure 4.1 with English and Finnish version of the same sentence. English uses a preposition to mark an oblique dependent while Finnish uses the elative case suffix instead. And the three definite articles in English have no counterpart on the Finnish side. Analytical languages like English use more words than synthetic languages like Finnish – in the example, the same meaning is expressed by 8 English words but only 4 Finnish words. If a parser makes one error in each language, its LAS will be 87.5% on English but only 75% on Finnish. One could object that more words also provide more opportunity to make an error; but it often seems to be the case that function words are easy to attach, making it easier for the parsers to reach higher scores on analytical languages. To be able to evaluate the impact of such language differences, we used additional evaluation metrics in the shared tasks. In 2017 the additional metric was CLAS [Nivre and Fang, 2017], which disregards attachment of function words in the total score. For the 2018 task I proposed MLAS [Zeman et al., 2018], which instead combines attachment of content words, attachment of function words and morphological features into

¹In the 2006 and 2007 tasks one would even expect gold-standard POS tags on input, so the evaluation of the parsing algorithm is not ‘biased’ by possible tagging errors, but by 2017 it was generally acknowledged that it is important to also evaluate parsing with machine-predicted tags—if the parser needs to see the tags at all.

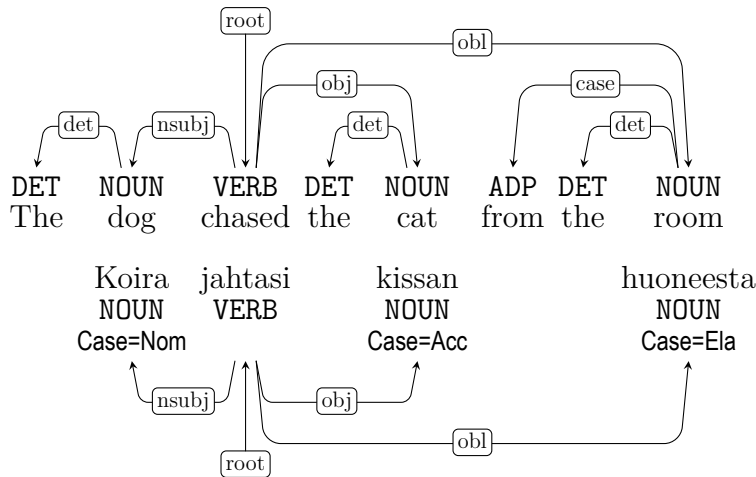


Figure 4.1: Impact of function words on parser evaluation. Adapted from Nivre and Fang [2017].

one score.² In the example in Figure 4.1, both English and Finnish have just 4 content words that can be correct or wrong, and to be correct the word must have its incoming dependency relation as well as all morphological features and all dependent function words analyzed correctly.

The shared task overview papers analyze the parsing results from many different angles. Here we just note that in the 2018 shared task, the best system’s LAS, macro-averaged over 61 ‘bigger’ datasets (those with large training data) reached 84%; the same figure for MLAS is 73%. The easiest dataset was one of the Polish treebanks (LAS 95%, MLAS 87%); the best result on Czech was LAS 92% and MLAS 85%; on Finnish it was LAS 90% and MLAS 84%; and on English LAS 88% and MLAS 76%. Low-resource languages obviously received much lower scores, especially under the stricter MLAS evaluation. Nine languages in the 2018 task were categorized as low-resource because they had either no labeled training data at all (Breton, Faroese, Naija, and Thai), or there was only a tiny sample of a few dozen sentences (Armenian, Buryat, Kazakh, Kurmanji, and Upper Sorbian). The average score on these languages achieved by the best system was 28% LAS but only 6% MLAS, showing that prediction of morphological features for an unknown language was still an extremely hard task. Nevertheless, there were significant differences among these languages. Some of them benefited from resource-rich siblings and ranked high above the low-resource average: Faroese (Germanic languages; LAS 49%, MLAS 1%), Upper Sorbian (Slavic languages; LAS 46%, MLAS 9%), Breton (Celtic languages; LAS 39%, MLAS 14%), and Armenian (Indo-European; LAS 37%, MLAS 13%).

²I also proposed a third metric, BLEX, which reflects syntax and lemmatization. All three metrics (LAS, MLAS, BLEX) were declared equally important – we wanted to encourage the participants to submit systems that predict all types of annotation.

5. Future Directions

After nine years of existence, the UD project is still growing and getting more diverse. New languages are added in every release,¹ new treebanks and genres are added to existing languages, annotated data is added to existing treebanks. Also growing is the community of researchers that contribute to UD and those that use it for their research. I am happy to be part of this endeavor and I hope it will keep growing for many years, as there are still hundreds of languages that lack digital resources. Nevertheless, morphosyntax is not the only area of language processing where annotated data are needed.

There are multiple proposals to either enhance the UD collection with new annotation layers, or to build other multilingual resources that are separate from UD but strive to follow a similar model of “universal” guidelines that would be applied to all languages. I will now discuss some of these new projects that I am involved in. Most of them revolve around getting closer to the semantics of natural language [Žabokrtský et al., 2020].

UD itself has always foreseen an optional second layer of annotation, called **enhanced representation** or **Enhanced Universal Dependencies (EUD)**. A similar layer existed already in Stanford Dependencies and the corresponding UD proposal was first presented by Schuster and Manning [2016]. EUD is a moderate attempt to make explicit some of the relations that are implicitly contained in the syntactic representation and that may be useful for language understanding applications. It is a deep syntactic layer but it does not aspire to provide a complete account of deep syntax (as opposed to other multi-layered syntactic frameworks, most notably the tectogrammatical layer of the Prague Dependency Treebank [Hajič et al., 2000]). Figure 5.1 exemplifies all major enhancements in EUD: 1. abstract nodes for predicates in gapping constructions (the verbs *chce* “wants” and *jet* “go”); 2. parent propagation across coordination (the second root relation to the abstract *chce*); 3. shared dependent of coordination (the second *advmod* relation to the adverb *ted* “now”); 4. grammatical coreference between the subjects of the control verb *chce* and the controlled infinitive *jet*; 5. grammatical coreference between the relative pronoun *nějž* “which” and its antecedent *kraje* “region”; and 6. relation labels enriched by case markers (*obl:do:gen*) and conjunction lemmas (*conj:a*). Note that the enhanced structure is a directed graph but it is no longer a tree.

Some of the enhancements can be derived almost deterministically from the basic dependency structure, others can be estimated with reasonable accuracy using language-specific heuristics. This has been suggested already by Nivre et al. [2018] and confirmed during two shared tasks in Enhanced UD parsing that I co-organized [Bouma et al., 2020, 2021]. In spite of it, only a fraction of the present UD treebanks² have the enhanced annotation layer. Ensuring that the other UD treebanks contain at least this minimal deep-syntactic representation is one research direction worth pursuing. However, I believe that we can also go deeper. The rather arbitrary selection of six enhancements can be extended in

¹UD releases occur regularly twice a year, in May and November.

²There are 32 treebanks of 17 different languages that have at least one of the six officially defined enhancements.

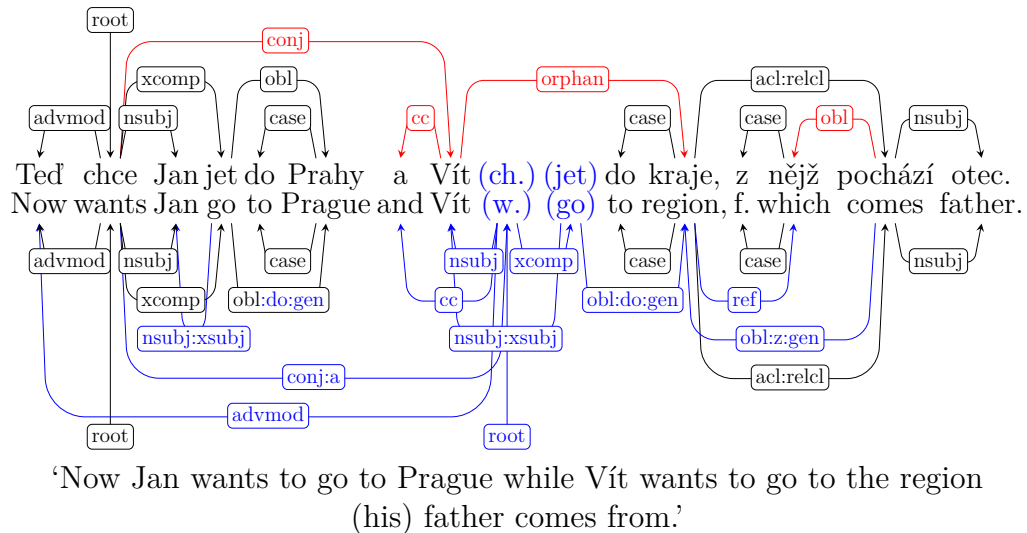


Figure 5.1: Example of basic UD tree (above the sentence) and corresponding enhanced UD graph (below). Colors highlight differences between the two structures.

the same spirit to constructions that are similar to those already covered by the guidelines, yet the guidelines do not mention them – sometimes perhaps because the constructions look different from English. For example, in languages such as Tamil the only way of creating a relative clause is a participle. Not only could the relative clause enhancement be extended to relative participles, it could also be extended to participial modifiers in English. Semi-automatic methods could be applied to normalize syntactic alternations [Candito et al., 2017] such as passives, antipassives, reflexives or causatives.

I outlined these ideas together with my PhD student Kira Droganova in Droganova and Zeman [2019] (Section 6.6); to distinguish the new extensions from the already defined Enhanced UD layer, we call it **Deep Universal Dependencies**. We envision a two-speed scenario. On one hand, we want to have cross-linguistically applicable guidelines for many different phenomena that exist between surface syntax and semantics and can be captured in annotated corpora. Conversion procedures could be defined to translate corresponding language resources to the ‘universal’ framework for languages for which such resources already exist. On the other hand, we are well aware that annotations of this kind are difficult and expensive to obtain, so we cannot hope for a growth rate comparable to Universal Dependencies. That is why semi-automatic approaches and heuristics are important, as we can use them to obtain less detailed and less accurate, but still useful annotation for a much larger set of languages. Kira is currently looking into a unified taxonomy of deep syntactic relations that would identify the common ground between several influential frameworks such as the tectogrammatical functors from PDT [Hajič et al., 2000], the PropBank roles [Palmer et al., 2005], or the MTT-inspired annotation of AnCora [Taulé et al., 2008].

Another large area is annotation of entities and **coreference** between them. Not just grammatical coreference, which is conditioned by syntax and which is at least partially covered by Enhanced UD, but all other **mentions** (by name, common noun, pronoun...) that can be said, based on context, to be representing the same entity. Delimitation of mentioning expressions is based on syntactic units,

which provides a potential link between Universal Dependencies and coreference annotation. There are coreference-annotated datasets for multiple languages, some of them with and others without syntax, but each following its own annotation scheme. My colleagues and I have thus launched a project called **CorefUD** [Nedoluzhko et al., 2022] where we collect such resources, convert them to a common format and combine them with UD-style morphosyntactic annotation. It currently contains 17 datasets of 12 languages. These datasets have been harmonized at the level of file format and a bit beyond, e.g. with regard to the set of entity types used. However, the common linguistic guidelines are yet to be defined: for example, how exactly should we delimit a mention given its syntactic environment? How do we capture ‘zero’ mentions that are reflected solely by agreement on the verb? Another PhD student supervised by me, Dima Taji, is just starting research along these lines.

The third multilingual project I want to mention is **Uniform Meaning Representation (UMR)** [Van Gysel et al., 2021]. This one really belongs to the level of semantics, rather than deep syntax. There is no effort at present to map it to syntactic frameworks such as UD, yet the meeting point is that both UD and UMR’s objective is to design structured annotation of sentences that would use the same set of concepts across all human languages. Pilot annotations already exist for six languages from six different families. With my colleagues from ÚFAL I am now investigating how UMR can be applied to other languages, primarily to Czech, and (together with my third PhD student Federica Gamba) to Latin.

The last two directions I want to mention here are back in the realm of morphosyntax. Both of them are potential extensions of UD annotation and both of them attempt to overcome problems that stem from taking the word as the basic unit of annotation. The two research directions try to loosen the impact of word boundaries and are complementary: One looks at small phrases, i.e., above the word level, the other looks at morphemes and other sub-word units, i.e., below the word level [Zeman, 2023].

The morphological features in Interset and in UD are defined for individual words; but in many languages, grammatical meanings such as tense and aspect are expressed analytically, using a content word in combination with one or more function words. For example, past perfect (pluperfect) in English is constructed using a finite past tense of the auxiliary *have* and the past participle of a content verb, as in *We had spoken*. None of the words involved is specific to pluperfect, and none of them will get the feature `Tense=Pqp` that encodes pluperfect. Therefore the annotation does not reveal that it is the same construction as Portuguese *Nós faláramos* – here the verb will be annotated as pluperfect, which is expressed purely morphologically. To facilitate such comparisons, we can define a new annotation layer in which UD-like features will be attributed to phrases, possibly discontinuous. So in Czech *Nejsem a nikdy jsem nebyl vázán touto smlouvou* “I am not and never have been bound by this contract”, we could say that the phrase *nejsem vázán* is finite indicative present tense passive, while *jsem nebyl vázán* is finite indicative past tense passive; note that both of them share the word *vázán*, which itself is only a passive participle (non-finite, with no tense feature).

On the other hand, dependency relations in UD are defined between words but not between smaller units. This is not ideal in certain use cases and certain languages. One cannot see parallel structure between compounds in English,

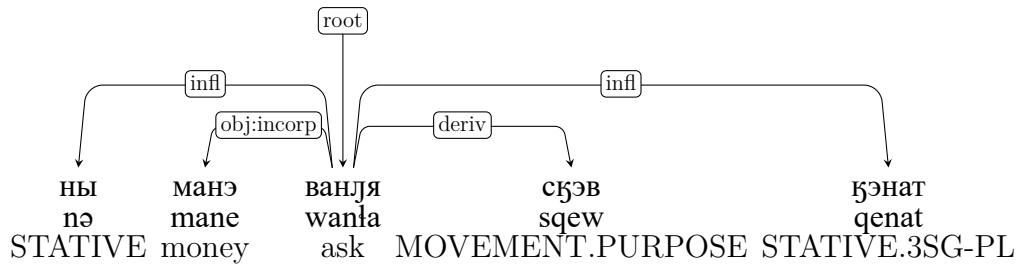


Figure 5.2: A dependency tree over the morphemes of the Chukchi word *нѡманѡванлѡсѡѡѡнѡт* (*nəmanəwanlasqewqenat*) “they constantly asked for money”.

where they are usually written as multiple words (*life insurance company*) and in German, where the same compound is typically written as one word (*Lebensversicherungsgesellschaft*). In other languages there are other reasons why a word may cover an entire sentence: agglutinating languages such as Turkish support long derivation chains (*çöplüklerimizdekilerdenmiydi* “was it from those that were in our garbage cans?”), polysynthetic languages like Chukchi may incorporate object of a verb inside the verb (*нѡманѡванлѡсѡѡѡнѡт* (*nəmanəwanlasqewqenat*) “they constantly asked for money” incorporates the object *манѡ* “money” in the verb). A syntactic tree of a sentence with one or two words will not reveal the structure and relations that exist inside the word. One can thus ask whether we can define a similar dependency structure over morphemes rather than words, or at least over sub-word units that have their own lexical content and may correspond to words in other languages. Such extensions have been proposed in the UD community [Tyers and Mishchenkova, 2020] (Figure 5.2) and similar ideas are also pursued by my colleagues at ÚFAL [Žabokrtský et al., 2022].

To summarize, Universal Dependencies and its predecessors have shown that there is a need for linguistically annotated data that cover many human languages and apply a unified annotation framework to all these languages. Almost 150 languages now have such resources at the level of segmentation, morphology and surface syntax, and these resources are widely used in natural language processing, linguistics and digital humanities in general. This effort can and should be extended to other languages, but also to other areas of natural language understanding, such as deep syntax and semantics.

6. Selected Publications

6.1 Cross-Language Parser Adaptation between Related Languages

Full reference: Daniel Zeman and Philip Resnik. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42, Hyderabad, India, January 2008. Asian Federation of Natural Language Processing. URL <https://aclanthology.org/I08-3008.pdf>. [Zeman and Resnik, 2008]

Comments: The term *delexicalized parsing* was coined in this paper. We presented experiments with transfer of parsing models from Danish to Swedish, where Swedish served as a surrogate for a low-resource language. Besides delexicalized parsing (Section 1.2), we also evaluated test data translation (Section 1.3), and found the former to perform better on our dataset. Our proposals were further developed and evaluated on multiple languages by McDonald et al. [2011], which sparked more interest by a number of other researchers. Nowadays, delexicalized parsing is still occasionally used as a cheap and quick first step for resourceless languages; however, lexicalized parsers using large multilingual language models typically perform better (even on languages not contained in their training data). My contribution: about 70%. Number of citations according to Google Scholar (retrieved 2023-07-21): **240**.

Cross-Language Parser Adaptation between Related Languages

Daniel Zeman

Univerzita Karlova
Ústav formální a aplikované lingvistiky
Malostranské náměstí 25
CZ-11800 Praha
zeman@ufal.mff.cuni.cz

Philip Resnik

University of Maryland
Department of Linguistics and
Institute for Advanced Computer Studies
College Park, MD 20742, USA
resnik@umd.edu

Abstract

The present paper describes an approach to adapting a parser to a new language. Presumably the target language is much poorer in linguistic resources than the source language. The technique has been tested on two European languages due to test data availability; however, it is easily applicable to any pair of sufficiently related languages, including some of the Indic language group. Our adaptation technique using existing annotations in the source language achieves performance equivalent to that obtained by training on 1546 trees in the target language.

1 Introduction

Natural language parsing is one of the key areas of natural language processing, and its output is used in numerous end-user applications, e.g. machine translation or question answering. Unfortunately, it is not easy to build a parser for a resource-poor language. Either a reasonably-sized syntactically annotated corpus (treebank) or a human-designed formal grammar is typically needed. These types of resources are costly to build, both in terms of time and of the expenses on qualified manpower. Both also require, in addition to the actual annotation process, a substantial effort on treebank/grammar design, format specifications, tailoring of annotation guidelines etc; the latter costs are rather constant no matter how small the resulting corpus is.

In this context, there is the intriguing question whether we can actually build a parser without a treebank (or a broad-coverage formal grammar) of the particular language. There is some related work that addresses the issue by a variety of means.

Klein and Manning (2004) use a hybrid unsupervised approach, which combines a constituency and a dependency model, and achieve an unlabeled F-score of 77.6% on Penn Treebank Wall Street Journal data (English), 63.9% on Negra Corpus (German), and 46.7% on the Penn Chinese Treebank.¹ Bod (2006) uses unsupervised data-oriented parsing; the input of his parser contains manually assigned gold-standard tags. He reports 64.2% unlabeled F-score on WSJ sentences up to 40 words long.²

Hwa et al. (2004) explore a different approach to attacking a new language. They train Collins's (1997) Model 2 parser on the Penn Treebank WSJ data and use it to parse the English side of a parallel corpus. The resulting parses are converted to dependencies, the dependencies are projected to a second language using automatically obtained word alignments as a bridge, and the resulting dependency trees cleaned up using a limited set of language-specific post-projection transformation rules. Finally a dependency parser for the target language is trained on this projected dependency treebank, and the accuracy of the parser is measured against a gold standard. Hwa et al. report dependency accuracy of 72.1 for Spanish, comparable to a rule-based commercial parser; accuracy on Chinese is 53.9%, the equivalent of a parser trained on roughly 2000 sentences of the Penn Chinese Treebank (sentences ≤ 40 words, average length 20.6).

¹ Note that in all these experiments they restrict themselves to sentences of 10 words or less.

² On sentences of ≤ 10 words, Bod achieves 78.5% for English (WSJ), 65.4% for German (Negra) and 46.7% for Chinese (CTB).

Our own approach is motivated by McClosky et al.’s (2006) reranking-and-self-training algorithm, used successfully in adapting a parser to a new domain. One can easily imagine viewing two dialects of a language or even two related languages as two domains of one “super-language” while the vocabulary will certainly differ (due to independently designed orthographies for the two languages), many morphological and syntactic properties may be shared. We trained Charniak and Johnson’s (2005) reranking parser on one language and applied it to another closely related language. In addition, we investigated the utility of large but unlabeled data in the target language, and of a large parallel corpus of the two languages.³

2 Corpora and Other Resources

The selection of our source and target languages was driven by the need for two closely related languages with associated treebanks. (In a real-world application we would not assume the existence of a target-language treebank, but one is needed here for evaluation.) Danish served as the source language and Swedish as target, since these languages are closely related and there are freely available treebanks for both.⁴

The Danish Dependency Treebank (Kromann et al. 2004) contains 5,190 sentences (94,386 tokens). The texts come from the Danish Parole Corpus (1998–2002, mixed domain). We split the data into 4,900 training and 290 test sentences, keeping the 276 not exceeding 40 words.

The Swedish treebank Talbanken05 (Nivre et al. 2006) contains 11,042 sentences (191,467 tokens). It was converted at Växjö from the much older Talbanken76 treebank, created at the Lund University. Again, the texts belong to mixed domains. We split the data to 10,700 training and 342 test sentences, out of which 317 do not exceed 40 words.

Both treebanks are dependency treebanks, while the Charniak-Johnson reranking parser works with phrase structures. For our experiments, we con-

verted the treebanks from dependencies to phrases, using the “flattest-possible” algorithm (Collins et al. 1999; algorithm 2 of Xia and Palmer 2001). The morphological annotation of the treebanks helped us to label the non-terminals. Although the Charniak’s parser can be taught a new inventory of labels, we found it easier to map head morpho-tags directly to Penn-Treebank-style non-terminals. Hence the parser can think it’s processing Penn Treebank data. The morphological annotation of the treebanks is further discussed in Section 4.

We also experimented with a large body of unannotated Swedish texts. Such data could theoretically be acquired by crawling the Web; here, however, we used the freely available JRC-Acquis corpus of EU legislation (Steinberger et al. 2006).⁵ The Acquis corpus is segmented at the paragraph level. We ran a simple procedure to split the paragraphs into sentences and pruned sentences with suspicious length, contents (sequence of dashes, for instance) or both. We ended up with 430,808 Swedish sentences and 6,154,663 tokens.

Since the Acquis texts are available in 21 languages, we can also exploit the Danish Acquis and its alignment with the Swedish one. We use it to study the similarity of the two languages, and for the “gloss” experiment in Section 5.1. Paragraph-level alignment is provided as part of Acquis and contains 283,509 aligned segments. Word-level alignment, needed for our experiment, was obtained using GIZA++ (Och and Ney 2000).

The treebanks are manually tagged with parts of speech and morphological information. For some of our experiments, we needed to automatically re-tag the target (Swedish) treebank, and to tag the Swedish Acquis. For that purpose we used the Swedish tagger of Jan Hajič, a variant of Hajič’s Czech tagger (Hajič 2004) retrained on Swedish data.

3 Treebank Normalization

The two treebanks were developed by different teams, using different annotation styles and guidelines. They would be systematically different even if their texts were in the same language, but it is

³ There are other approaches to domain adaptation as well. For instance, Steedman et al. (2003) address domain adaptation using a weakly supervised method called co-training. Two parsers, each applying a different strategy, mutually prepare new training examples for each other. We have not tested co-training for cross-language adaptation.

⁴ We used the CoNLL 2006 versions of these treebanks.

⁵ Legislative texts are a specialized domain that cannot be expected to match the domain of our treebanks, however vaguely defined it is. But presumably the domain matching would be even less trustworthy if we acquired the unlabeled data from the web.

the impact of the language difference, not annotation style differences, that we want to measure; therefore we normalize the treebanks so that they are as similar as possible.

While this may sound suspicious at first glance (“wow, are they refining their test data?!”), it is important to understand why it does not unacceptably bias the results. If our method were applied to a new language, where no treebank exists, trees conforming to the annotation scenario of a treebank of related language would be perfectly satisfying. In addition, note that we apply only systematic changes, mostly reversible. Moreover, the transformations can be done on the training data side, instead of test data.

Following are examples of the style differences that underwent normalization:

DET-ADJ-NOUN. Da: *de norske piger*. Sv:⁶ *en gammal institution* (“an old institution”) In DDT, the determiner governs the adjective and the noun. The approach of Talbanken (and of a number of other dependency treebanks) is that both determiner and adjective depend on the noun.

NUM-NOUN. Da: *100 procent* (“100 percent”) Sv: *två eventuellt tre år* (“two, possibly three years”) In DDT, the number governs the noun. In Talbanken, the number depends on the noun.

GENITIVE-NOMINATIVE. Da: *Ruslands vej* (“Russia’s way”) Sv: *års inkomster* (“year’s income”). In DDT, the nominative noun (the owned) governs the noun in genitive (the owner). Talbanken goes the opposite way.

COORDINATION. Da: *Færøerne og Grønland* (“Faroe Islands and Greenland”) Sv: *socialgrupper, nationer och raser* (“social groups, nations and races”) In DDT, the last coordination member depends on the conjunction, the conjunction and everything else (punctuation, inner members) depend on the first member, which is the head of the coordination. In Talbanken, every member depends on the previous member, commas and conjunctions depend on the member following them.

4 Mapping Tag Sets

The nodes (words) of the Danish Dependency Treebank are tagged with the Parole morphological

tags. Talbanken is tagged using the much coarser Mamba tag set (part of speech, no morphology). The tag inventory of Hajič’s tagger is quite similar to the Danish Parole tags, but not identical. We need to be able to map tags from one set to the other. In addition, we also convert pre-terminal tags to the Penn Treebank tag set when converting dependencies to constituents.

Mapping tag sets to each other is obviously an information-lossy process, unless both tag sets cover identical feature-value spaces. Apart from that, there are numerous considerations that make any such conversion difficult, especially when the target tags have been designed for a different language.

We take an Interlingua-like (or Inter-tag-set) approach. Every tag set has a *driver* that implements decoding of the tags into a nearly universal feature space that we have defined, and encoding of the feature values by the tags. The encoding is (or aims at being) independent of where the feature values come from, and the decoding does not make any assumptions about the subsequent encoding. Hence the effort put in implementing the drivers is reusable for other tagset pairs.

The key function, responsible for the universality of the method, is `encode()`. Consider the following example. There are two features set, POS = “noun” and GENDER = “masc”. The target set is not capable of encoding masculine nouns. However, it allows for “noun” + “com” | “neut”, or “pronoun” + “masc” | “fem” | “com” | “neut”. An internal rule of `encode()` indicates that the POS feature has higher priority than the GENDER feature. Therefore the algorithm will narrow the tag selection to noun tags. Then the gender will be forced to common (i.e. “com”).

Even the precise feature mapping does not guarantee that the *distribution* of the tags in two corpora will be reasonably close. All converted source tags will now fit in the target tag set. However, some tags of the target tag set may not be used, although they are quite frequent in the corpus where the target tags are native. Some examples:

- Unlike in Talbanken, there are no **determiners** in DDT. That does not mean there are no determiners in Danish – but DDT tags them as pronouns.

⁶ These are separate examples from the two treebanks. They are *not* translations of each other!

Bestemmelserne	i denne aftale kan	ændres	og	revideres	helt eller delvis efter	fælles
Bestämmelserna	i detta avtal får	ändras	eller	revideras	helt eller delvis efter	gemensam
överenskomst	mellem parterne.					
överenskommelse	mellan parterna.					

Figure 1. Comparison of matching Danish (upper) and Swedish (lower) sentences from Acquis. Despite the one-to-one word mapping, only the 5 bold words have identical spelling.

- Swedish tags encode a special feature of **personal pronouns**, “subject” vs. “object” form (the distinction between English *he* and *him*). DDT calls the same paradigm “nominative” vs. “unmarked” case.
- Most noun phrases in both languages distinguish just the **common and neuter genders**. However, some pronouns could be classified as masculine or feminine. Swedish tags use the masculine gender, Danish do not.
- DDT does not use special part of speech for **numbers** — they are tagged as adjectives.

All of the above discrepancies are caused by differing designs, not by differences in language. The only linguistically grounded difference we were able to identify is the **supine** verb form in Swedish, missing from Danish.

When not just the tag *inventories*, but also the tag *distributions* have to be made compatible (which is the case of our delexicalization experiments later in this paper), we can create a new *hybrid* tag set, omitting any information specific for one or the other side. Tags of both languages can then be converted to this new set, using the universal approach described above.

5 Using Related Languages

The Figure 1 gives an example of matching Danish and Swedish sentences. This is a real example from the Acquis corpus. Even a non-speaker of these languages can detect the evident correspondence of at least 13 words, out of the total of 16 (ignoring final punctuation). However, due to different spelling rules, only 5 word pairs are string-wise identical. From a parser’s perspective, the rest is unknown words, as it cannot be matched against the vocabulary learned from training data.

We explore two techniques of making unknown words known. We call them *glosses* and *delexicalization*, respectively.

5.1 Glosses

This approach needs a Danish-Swedish (da-sv) bitext. As shown by Resnik and Smith (2003), parallel texts can be acquired from the Web, which makes this type of resource more easily available than a treebank. We benefited from the Acquis da-sv alignments.

Similarly to phrase-based translation systems, we used GIZA++ (Och and Ney 2000) to obtain one-to-many word alignments in both directions, then combined them into a single set of refined alignments using the “final-and” method of Koehn et al. (2003). The refined alignments provided us with two-way tables of a source word and all its possible translations, with weights. Using these tables, we glossed each Swedish word by its Danish, using the translation with the highest weight.

The glosses are used to replace Swedish words in test data by Danish, making it more likely that the parser knows them. After a parse has been obtained, the trees are “restuffed” with the original Swedish words, and evaluated.

5.2 Delexicalization

A second approach relies on the hypothesis that the interaction between morphology and syntax in the two languages will be very similar. The basic idea is as follows: Replace Danish words in training data with their morphological (POS) tags. Similarly, replace the Swedish words in test data with tags. This replacement is called delexicalization. Note that there are now two levels of tags in the trees: the Danish/Swedish tags in terminal nodes, and the Penn-style tags as pre-terminals. The terminal tags are more descriptive because both Nor-

dic languages have a slightly richer morphology than English, and the conversion to the Penn tag set loses information.

The crucial point is that both Danish and Swedish use the same tag set, which helps to deal with the discrepancy between the training and the test terminals.

Otherwise, the algorithm is similar to that of glosses: train the parser on delexicalized Danish, run it over delexicalized Swedish, restuff the resulting trees with the original Swedish words (“re-lexicalize”) and evaluate them.

6 Experiments: Part One

We ran most experiments twice: once with Charniak’s parser alone (“C”) and once with the reranking parser of Charniak and Johnson, which we label simply Brown parser (“B”).

We use the standard `evalb` program by Sekine and Collins to evaluate the parse trees. Keeping with tradition, we report the F-score of the *labeled* precision and recall on the sentences of up to 40 words.⁷

Language	Parser	P	R	F
da	C	77.84	78.48	78.16
	B	78.28	78.20	78.24
da-hybrid	C	79.50	79.73	79.62
	B	80.60	79.80	80.20
sv	C	77.61	78.00	77.81
	B	79.16	78.33	78.74
sv-mamba	C	77.54	78.93	78.23
	B	79.67	79.26	79.46
sv-hybrid	C	76.10	76.04	76.07
	B	78.12	75.93	77.01

Table 1. Monolingual parsing accuracy.

To put the experiments in the right context, we first ran two monolingual tracks and evaluated Danish-trained parsers on Danish, and Swedish-trained parsers on Swedish test data. Both treebanks have also been parsed after delexicalization into various tag sets: Danish gold standard converted to the hybrid sv/da tag set, Swedish Mamba gold standard, and Swedish automatically tagged with hybrid tags.

The reranker did not prove useful for lexicalized Swedish, although it helped with Danish. (We cur-

⁷ $F = 2 \times P \times R / (P + R)$

rently have no explanation of this.) On the other hand, delexicalized reranking parsers outperformed lexicalized parsers for both languages. This holds for delexicalization using the gold standard tags (even though the Mamba tag set encodes much less information than the hybrid tags). Automatically assigned tags perform significantly worse.

Our baseline condition is simply to train the parsers on Danish treebank and run them over Swedish test data. Then we evaluate the two algorithms described in the previous section: glosses and delexicalization (hybrid tags).

Approach	Parser	P	R	F
baseline	C	44.59	42.04	43.28
	B	42.94	40.80	41.84
glosses	C	61.85	65.03	63.40
	B	60.22	62.85	61.50
delex	C	63.47	67.67	65.50
	B	64.74	68.15	66.40

Table 2. Cross-language parsing accuracy.

7 Self-Training

Finally, we explored the self-training based domain-adaptation technique of McClosky et al. (2006) in this setting. McClosky et al. trained the Brown parser on one domain of English (WSJ), parsed a large corpus of a second domain (NANTC), trained a new Charniak (non-reranking) parser on WSJ plus the parsed NANTC, and tested the new parser on data from a third domain (Brown Corpus). They observed improvement over baseline in spite of the fact that the large corpus was not in the third domain.

Our setting is similar. We train the Brown parser on Danish treebank and apply it to Swedish Acquis. Then we train new Charniak parser on Danish treebank *and* the parsed Swedish Acquis, and test the parser on the Swedish test data. The hope is that the parser will get lexical context for the structures from the parsed Swedish Acquis.

We did not retrain the reranker on the parsed Acquis, as we found it prohibitively expensive in both time and space. Instead, we created a new Brown parser by combining the new Charniak parser, and the old reranker trained only on Danish.

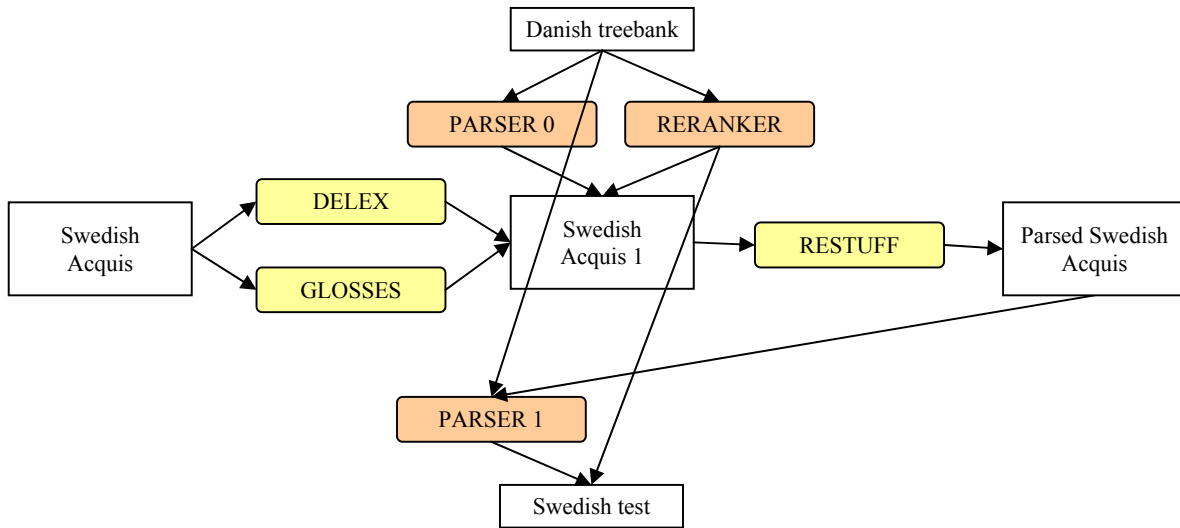


Figure 2. Scheme of the self-training system.

A different scenario is used with the gloss and delex techniques. In this case, we only use delexicalization/glosses to parse the Acquis corpus. The new Charniak model is always trained directly on lexicalized Swedish, i.e. the parsed Acquis is restuffed before being handed over to the trainer. Table-3 shows the corresponding application chart.

8 Experiments: Part Two

The following table shows the results of the self-training experiments. All F-scores outperform the corresponding results obtained without self-training.

Approach	Parser	P	R	F
Plain	C	45.14	43.96	44.54
	B	43.12	42.23	42.67
Glosses	C	62.87	66.17	64.48
	B	61.94	64.77	63.32
Delex	C	55.87	63.86	59.60
	B	53.87	61.45	57.41

Table 3. Self-training adaptation results.

Not surprisingly, the Danish-trained reranker does not help here. However, even the first-stage parser failed to outperform the Part One results. Therefore the 66.40% labeled F-score of the delexicalized Brown parser is our best result. It im-

proves the baseline by 23% absolute, or 41% error reduction.

9 Discussion

As one way of assessing the usefulness of the result, we compared it to the learning curve on the Swedish treebank. This corresponds to the question “How big a treebank would we have to build, so that the parser trained on the treebank achieves the same F-score?” We measured the F-scores for Swedish-trained parsers on gradually increasing amounts of training data (50, 100, 250, 500, 1000, 2500, 5000 and 10700 sentences).

The learning curve is shown in Figure 3. Using interpolation, we see that more than 1500 Swedish parse trees would be required for training, in order to achieve the performance we obtained by adapting an existing Danish treebank. This result is similar in spirit to the results Hwa et al. (2004) report when training a Chinese parser using dependency trees projected from English. As they observe, creating a treebank of even a few thousand trees is a daunting undertaking – consistent annotation typically requires careful design of guidelines for the annotators, testing of the guidelines on data, refinement of those guidelines, ramp-up of annotators, double-annotation for quality control, and so forth. As a case in point, the Prague Dependency Treebank (Böhmová et al, 2003) project began in

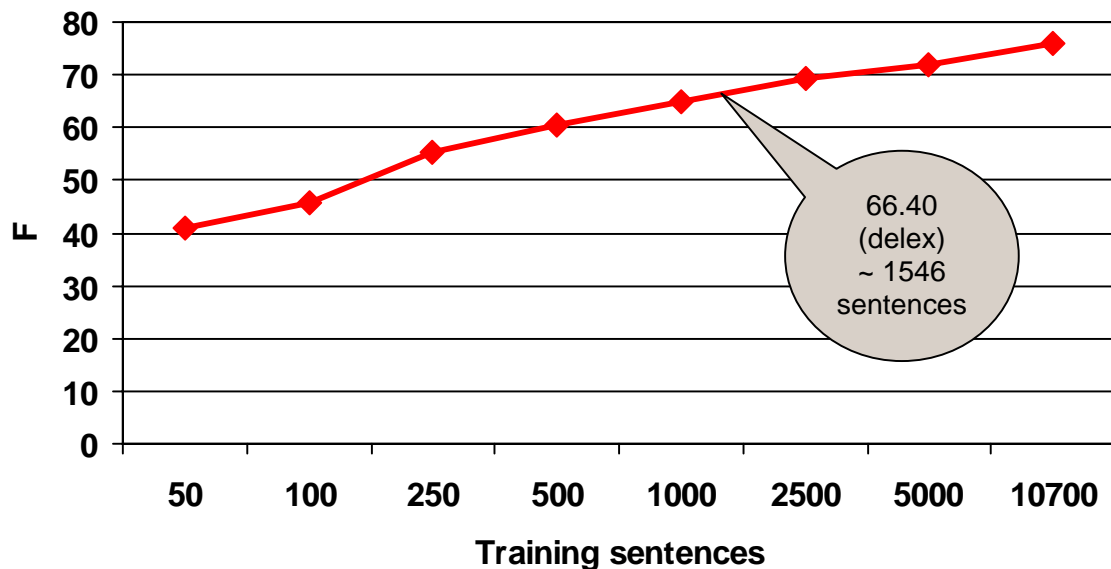


Figure 3. The learning curve on the Swedish training data.

1996, and required almost a year for its first 1000 sentences to appear (although things sped up quickly, and over 20000 sentences were available by fall 1998). In contrast, if the source and target language are sufficiently related – consider Danish and Swedish, as we have done, or Hindi and Urdu – our approach should in principle permit a parser to be constructed in a matter of days.).

9.1 Ways to Improve: Future Work

The 77.01% F-score of a parser trained on delexicalized automatically assigned hybrid Swedish tags is an upper bound. Some obvious ways of getting closer to it include better treebank and tag-set mapping and better tagging. In addition, we are interested in seeing to what extent performance can be further improved by better iterative self-training.

We also want to explore classifier combination techniques on glosses, delexicalization, and the N-best outputs of the Charniak parser. One could also go further, and explore a combination of techniques, e.g. taking advantage of the ideas proposed here in tandem with unsupervised parsing (as in Bod 2006) or projection of annotations across a parallel corpus (as in Hwa et al. 2004).

Acknowledgements

The authors thank Eugene Charniak and Mark Johnson for making their reranking parser available, as well as the creators of the corpora used in this research. We also thank the anonymous reviewers for useful remarks on where to focus our workshop presentation.

The research reported on in this paper has been supported by the Fulbright-Masaryk Fellowship (first author), and by Grant No. N00014-01-1-0685 ONR. Ongoing research (first author) is supported by the Ministry of Education of the Czech Republic, project MSM0021620838, and Czech Academy of Sciences, project No. 1ET101470416.

References

- Rens Bod. 2006a. *Unsupervised Parsing with U-DOP*. In: Proceedings of the Conference on Natural Language Learning (CoNLL-2006). New York, New York, USA.
- Rens Bod. 2006b. *An All-Subtrees Approach to Unsupervised Parsing*. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL (COLING-ACL-2006). Sydney, Australia.

- Alena Böhmová, Jan Hajič, Eva Hajičová, Barbora Hladká. 2003. *The Prague Dependency Treebank: A Three-Level Annotation Scenario*. In: Anne Abeillé (ed.): *Treebanks: Building and Using Syntactically Annotated Corpora*. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Eugene Charniak, Mark Johnson. 2005. *Coarse-to-Fine N-Best Parsing and MaxEnt Discriminative Reranking*. In: Proceedings of the 43rd Annual Meeting of the ACL (ACL-2005), pp. 173–180. Ann Arbor, Michigan, USA.
- Michael Collins. 1997. *Three Generative, Lexicalized Models for Statistical Parsing*. In: Proceedings of the 35th Annual Meeting of the ACL, pp. 16–23. Madrid, Spain.
- Michael Collins, Jan Hajič, Lance Ramshaw, Christoph Tillmann. 1999. *A Statistical Parser for Czech*. In: Proceedings of the 37th Annual Meeting of the ACL (ACL-1999), pp. 505–512. College Park, Maryland, USA.
- Jan Hajič. 2004. *Disambiguation of Rich Inflection (Computational Morphology of Czech)*. Karolinum, Charles University Press, Praha, Czechia.
- Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, Okan Kolak. 2004. *Bootstrapping Parsers via Syntactic Projection across Parallel Texts*. In: *Natural Language Engineering 1 (1)*: 1–15. Cambridge University Press, Cambridge, England.
- Dan Klein, Christopher D. Manning. 2004. *Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency*. In: Proceedings of the 42nd Annual Meeting of the ACL (ACL-2004). Barcelona, Spain.
- Philipp Koehn, Franz Josef Och, Daniel Marcu. 2003. *Statistical Phrase-Based Translation*. In: Proceedings of HLT-NAACL 2003, pp. 127–133. Edmonton, Canada.
- Matthias T. Kromann, Line Mikkelsen, Stine Kern Lynge. 2004. *Danish Dependency Treebank*. At: <http://www.id.cbs.dk/~mtk/treebank/>. København, Denmark.
- Mitchell P. Marcus, Beatrice Santorini, Mary Ann Minkiewicz. 1993. *Building a Large Annotated Corpus of English: the Penn Treebank*. In: *Computational Linguistics*, vol. 19, pp. 313–330.
- David McClosky, Eugene Charniak, Mark Johnson. 2006. *Reranking and Self-Training for Parser Adaptation*. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL (COLING-ACL-2006). Sydney, Australia.
- Joakim Nivre, Jens Nilsson, Johan Hall. 2006. *Talbanken05: A Swedish Treebank with Phrase Structure and Dependency Annotation*. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006). May 24–26. Genova, Italy.
- Franz Josef Och, Hermann Ney. 2000. *Improved Statistical Alignment Models*. In: Proceedings of the 38th Annual Meeting of the ACL (ACL-2000), pp. 440–447. Hong Kong, China.
- Philip Resnik, Noah A. Smith. 2003. *The Web as a Parallel Corpus*. In: *Computational Linguistics*, 29(3), pp. 349–380.
- Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlén, Steven Baker, Jeremiah Crim. 2003. *Bootstrapping Statistical Parsers from Small Datasets*. In: Proceedings of the 11th Conference of the European Chapter of the ACL (EACL-2003). Budapest, Hungary.
- Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufiş, Dániel Varga. 2006. *The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages*. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006). May 24–26. Genova, Italy.
- Fei Xia, Martha Palmer. 2001. *Converting Dependency Structures to Phrase Structures*. In: Proceedings of the 1st Human Language Technology Conference (HLT-2001). San Diego, California, USA.

6.2 Reusable Tagset Conversion Using Tagset Drivers

Full reference: Daniel Zeman. Reusable tagset conversion using tagset drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 213–218, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/66_paper.pdf. [Zeman, 2008]

Comments: This is the first and main reference for Interset (Chapter 2). A preliminary version of the tagset conversion system was used already in [Zeman and Resnik, 2008]. Besides being used to convert tags between existing tagsets, Interset gradually became a framework that could be used to describe and access word features in any language. It became part of the language-processing framework Treex [Popel and Žabokrtský, 2010],¹ it was extensively used in the HamleDT project (Section 6.3), and finally, selected features from Interset provided the morphological annotation layer in Universal Dependencies (Section 6.4). I continue to oversee and maintain the set of features documented in UD, as I did previously for Interset; I also keep the conversion libraries in sync with newly added features. Furthermore, my experience with morphosyntactic harmonization has projected into my monograph on the topic [Zeman, 2018]. My contribution: 100%. Number of citations according to Google Scholar (retrieved 2023-07-21): **209**.

¹<https://ufal.mff.cuni.cz/treex>

Reusable Tagset Conversion Using Tagset Drivers

Daniel Zeman

Univerzita Karlova

Ústav formální a aplikované lingvistiky

Malostranské náměstí 25

CZ-11800 Praha

E-mail: zeman@ufal.mff.cuni.cz

Abstract

Part-of-speech or morphological tags are important means of annotation in a vast number of corpora. However, different sets of tags are used in different corpora, even for the same language. Tagset conversion is difficult, and solutions tend to be tailored to a particular pair of tagsets. We propose a universal approach that makes the conversion tools reusable. We also provide an indirect evaluation in the context of a parsing task.

1. Introduction

Most annotated corpora use various types of tags to encode additional information on words. In some cases this information is merely the part of speech (“noun”, “verb” etc.—hence the term *part-of-speech* or *POS tags*). In many cases, however, the string of characters comprising the tag is a compressed representation of a feature-value structure. Most of the features encoded this way are morphosyntactic (e.g. “gender = masculine”, “number = singular”), hence the term *morphological tags*.

Unfortunately, it is very rare to see two corpora sharing a common set of tags. Language differences are only partially responsible—it is the corpus designers, their diverse views, theories and intended uses of the corpora, what matters most. Even two corpora of the same language may define two completely incompatible tagsets.

Such diversity proves disadvantageous for both human users and NLP software. A human user (linguist) typically wants to submit queries such as “show me all occurrences of a noun in plural, preceded by a preposition”. Tags however rarely contain statements like “number = plural” literally. That would be prohibitively space-consuming. Instead we have to know that e.g. the fourth character of the tag being “P” means “plural”. For instance, the tag NNIS7-----A-----¹ may read as “part of speech = noun, detailed part of speech = common noun, gender = masculine inanimate, number = singular, case = 7th (instrumental), negativeness = affirmative”. To work with the corpus efficiently, a linguist either needs to interpret the tags using specialized software, or to memorize the particular tag scheme. Obviously, if the same linguist has to switch to a different corpus, he/she must memorize more schemes or replace the tag interpretation software.

Similarly, various NLP tools may depend on particular tagsets. While some tools indeed treat tags as atomic strings, others could exploit the tag structure to dig more

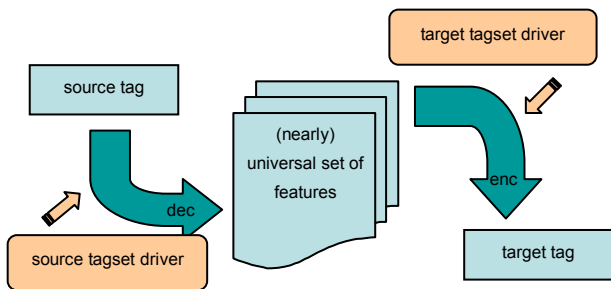
information about the word—no matter whether they use the features in machine learning, or in human-designed rules. If the tagset changes, manual rules become useless and statistical models have to be retrained at least; even that may not be possible in case the training procedure works with selected subsets of the feature pool. Applicability of NLP software to multiple corpora is exactly the reason why one would want to convert tags from one tagset to another.

For many tagset pairs, designing the conversion procedure is not easy. On one hand, there are rare tagsets (e.g. MULTEXT-EAST, Erjavec 2004) fitting at the same time languages as distant as Czech and Estonian; on the other hand, tagsets of two closely related languages (e.g. Danish and Swedish) or even two tagsets of the same language may differ substantially (for instance, the Mamba tagset of Swedish (Nivre et al. 2006) contains detailed classification of auxiliary verbs and punctuation but lacks features like number, mood, tense etc.; this is in sharp contrast to another Swedish tagset, Parole (Cinková and Pomikálek 2006), which in turn is not compatible with the Danish Parole (Kromann et al. 2004) tagset (the former classifies participles as verb forms, the latter as adjective forms; the former has separate tags for numerals, the latter classifies both cardinal and ordinal numbers as adjectives; etc.)

From the above said it follows that the typical tag conversion is an information-losing process. Though it is often desirable to perform it anyway and preserve as much information as possible. We have not been able to identify any previously published universal approach to do tagset conversion, which is not so surprising given the fact that for most part the conversion code must simply mimic the interpretation charts of the particular tagsets. We believe that most researchers solve the problem using specialized programs tailored to the two tagsets at hand. For subtly differing tagsets this may be the best thing to do; however, in all other cases, there is considerable effort put into analyzing the tag schemes, that cannot be reused for converting, say, the same source tagset into a new target tagset.

¹ This example is taken from the Prague Dependency Treebank (Böhmová et al. 2003)

In the present paper we propose an approach that makes the conversion code reusable. We define a (nearly) universal set of features and their values, and describe a way how *tagset drivers* can be used to convert various



tagsets in and out of the universal feature set. In Section 2 we describe our universal set of features, in Section 3 we describe the encoding algorithm and the architecture of tagset drivers, in Section 4 we mention difficult phenomena and in Section 5 we present experiments.

2. Universal Set of Features

The key idea of our approach is to have a feature structure capable of storing all or most information from any tagset. The structure contains all features whose values are usually encoded in tags. The role of this universal set (“*Interaset*”) is similar to the role of Interlingua in Interlingua-based machine translation (Richens 1958) or the role of Unicode among character sets. The *Interaset* serves as an intermediate step on the way from tagset A to tagset B. The interaction between the *Interaset* and tagsets A and B, respectively, is described in what we call *tagset drivers*. Once we write the drivers, we can do the two-way conversion A to B and B to A, plus the conversion between one of these tagsets and any other tagset that has been defined so far.

We are not likely to spare much time during the initial phase, if compared to just writing a targeted A-to-B conversion procedure. Actually, covering two completely new tagsets requires more work and care: we should describe both encoding and decoding of each tagset, we may have to think about features that are present in neither of them, and we will probably want to be more careful about aspects that may not matter to our current application. However, the reusability of the resulting code should compensate for the effort more than adequately. Plus we provide some algorithms to make adding new tagsets easier, and it is also possible that the required tagset has been covered by someone else who is sharing the code on the web.

Having analyzed about dozen tagsets,² we have identified the following features:

- part of speech
- various features for further details on part-of-speech: subpos, pronoun type, punctuation class and side (left vs. right bracket), syntactic part of speech, subcat
- yes/no features related to part of speech: possessive, reflexive, foreign, abbreviation, first part of hyphenated compound
- various inflectional features: gender, animateness, number, case, degree, definiteness, negativeness, person, politeness, possessor’s gender and number, verbal form, mood, tense, subtense, aspect, voice
- the rest: style, variant, other, tagset

Although covering new tagsets may lead to adding new features to the central pool, it is desirable to find most of them in the very beginning. It is good to know what can be there when writing drivers. On the other hand, we do not intend to cram the *Interaset* with hundreds of features, each of them specific to just one corpus. Some information in tags is really difficult to use out of the context of the original tagset. It is delicate to judge what belongs here; however, if there were a tag defined as “the word ‘apple’ occurring in a nested clause,” we could probably live without that information saved. The only reason of saving really everything is that converting a tagset to itself should not lose information. For that purpose we use the “other” feature. It contains arbitrary information that does not fit in other features and distinguishes tags. Since the information is not understood by any other tagset, we need to know which tagset the value comes from. Thus the identifier of the tagset should be stored in the “tagset” feature.

Except for “tagset” and “other”, there is a predefined list of possible values for each feature. Every feature also allows the empty value. While several feature-based tagsets distinguish between unknown values and irrelevant features, we do not find it wise in *Interaset*. For instance, the fifth character in the PDT Czech tagset identifies grammatical case. Its normal values are 1 to 7. For parts of speech that do not have case (e.g. interjections) the fifth character is – (dash). Adjectives generally do have case, yet there are borrowed words without Czech case suffixes whose case value is unknown (X). An example is the tag AAIPX----1A---- for “Buenos” in *Buenos Aires*. The benefit of making this distinction explicit in a tagset is unclear. What is clear, however, is that we must not reflect it in the universal feature set. Who can say that a feature will be irrelevant—given the context of the values of the other features—in any tagset whatsoever? It is quite easy to find features that are relevant in one tagset and not the other: e.g. Czech past participles distinguish gender, English don’t.

3. Tagset Drivers

While the *Interaset* is merely an abstract definition, the real implementation lies in the tagset drivers. A driver is a code library responsible for decoding and encoding tags.

² Penn tagset of English, PDT tagset of Czech, STTS tagset of German, Mamba and Parole tagsets of Swedish, CoNLL tagsets of Arabic, Bulgarian, Chinese, Danish (and of Czech, English, German and Swedish; these four are however based on the other tagsets mentioned earlier).

Decoding is reading a string (tag) into an internal data structure, in accordance with the list of possible features and their values. Encoding works the other way around.

The encoder obviously is the more difficult part. The decoder just reads and sorts the information, ideally not losing a single piece of it. If anything has to be discarded because it does not fit the target tagset, the discarding is encoder's task. There are two main reasons why encoding is not easy:

1. The encoder should be prepared to all values of all features, regardless that some of them are unknown in the particular tagset. For instance, if number = dual and the tagset does not know dual, it is probably better to encode plural than just leave number unknown.
2. Even if the target tagset knows features A and B, concrete value of A can restrict permitted values of B. Some *combinations* of feature values are not allowed. For instance, the Swedish Parole tagset allows "pos = noun & gender = common | neuter", and also "pos = pronoun & gender = masculine | feminine | common | neuter". If we are to encode "pos = noun & gender = masculine", we can either honor the part of speech, or the gender, but not both.

Fortunately enough, unknown feature values / combinations can be dealt with automatically if the driver has the list of all possible tags. By decoding all tags on the list, we get feature values for every tag. We thus know all feature values permitted in the given tagset and we know all value combinations. We have defined an ordered list of back-off values for every Interset feature value. The back-off lists contain all other values of the feature, including the empty value, so it is guaranteed that we always find a value that is permitted.³ Of course, the encoder can override the default back-off list if necessary.

As for unknown feature combinations, there is a predefined total ordering of the features that defines their priority (this can be overridden, too). Since features are ordered, all value combinations can be stored in a trie structure. On selecting value of a higher-priority feature, the structure immediately reveals restricted value space for all lower-priority features.

This back-off technique is implemented in a helper module. Any driver can call it and have the features adjusted to something the driver itself might produce during decoding. The encoder can then concentrate on the driver's native feature combinations. Besides that, the helper module can also check a driver's integrity by looking whether the decoder only sets known features and values, whether `encode(decode(x)) = x` etc.

The whole thing is implemented in Perl⁴. The drivers are Perl modules whose `encode` and `decode` functions can be called from other Perl programs, either to access

the feature values, or to convert tagsets. The conversion script is very simple and looks like this:⁵

```
use tagset::cs::pdt;
use tagset::en::penn;
while (<>)
{
    print tagset::en::penn::encode
        tagset::cs::pdt::decode $_, "\n";
}
```

So far we have implemented and tested drivers for several tagsets of the CoNLL 2006 (Buchholz and Marsi 2006) and 2007 (Nivre et al. 2007) shared task treebanks, for the Penn Treebank (Marcus et al. 1993), the Prague Dependency Treebank (Böhmová et al. 2003) and others, totaling 14 drivers. Those drivers are freely available on the web.⁶ We believe that the reusability will only be truly exploited if the drivers are shared in the community and we encourage everyone to contribute with drivers they need to write for themselves.

4. Difficult Phenomena

Working with various tagsets, we identified several fields that were difficult to capture and unify.

Endemic word classes were one example. Whenever seen fit, we tried to roof them with some more common parts of speech, instead of introducing a new high-level class. We wanted to reduce the necessity of encoders' taking care of parts of speech unknown in their home tagsets. Roofed word classes are usually distinguishable by one of the detailed-part-of-speech features.

Determiners, predeterminers and articles are one group of word classes missing in a substantial number of tagsets. We chose adjectives to serve as the roof class here. To pick another set of examples, here is an overview of various sorts of particles found in our tagsets:

- o unclassified particle (Czech TT, English RP, Swedish Q-----)
- o interrogative particle (Arabic FI هـ *hal*, Bulgarian Tn *ли li*)
- o affirmative particle (Bulgarian Ta *да da*)
- o negative particle (Arabic FN لا *lā*, Bulgarian Tn *не ne*, German PTKNEG *nicht*)
- o response particle (German PTKANT *ja* = "yes", *nein* = "no", *doch* = "yes", *danke* = "thank you"...))
- o auxiliary particle (Bulgarian Tx *да da* = "to", *ще šte* = "will")
- o modal particle (Bulgarian Tm *май maj* = "possibly")
- o verbal particle (Bulgarian Tv *нека neka* = "let")
- o emphasis particle (Bulgarian Te *даже даže* = "even")
- o gradable particle (Bulgarian Tg *най naj* = "most")

³ The necessary condition is that the decoder only sets known feature values, which is desirable anyway.

⁴ <http://www.perl.org/>

⁵ Real conversion script would also have to deal with the format in which the tags are mixed with text in the corpus. This example merely assumes a list of tags, without the actual words and other annotation.

⁶ <http://wiki.ufal.ms.mff.cuni.cz/user:zeman:interset>

- unique POS (Danish U, covering the words *at* = infinitival “to”, *som*, *der*)
- infinitive mark (German PTKZU *zu*, Swedish IM *att*, English TO *to* – includes prepositional occurrences of *to*)
- separated verbal prefix (German PTKVZ, *vor* in *stellen Sie sich vor*)
- adjectival particle (German PTKA, *am* in *am besten*, *zu* in *zu groß*)
- existential *there* in English (EX)
- measure word, quantifier (Chinese DM)
- genitive particle *de* in Chinese (DE 的 and 得)
- Chinese particles 了 *le* (perfect), 著 *zhe*, 起 *qǐ*, 過 *guò* (Di)
- Chinese particles 了 *le*, 的 *de*, 來 *lái* (Ta)
- Chinese particles 而已 *éryǐ*, 沒有 *méiyǒu*, 也罷 *yěba*, 沒有 *méiyǒu*, 好了 *hǎole* (Tb)
- Chinese particles 呢 *ne*, 吧 *ba*, 啊 *'a*, 囉 *luō* (Tc)
- Chinese particles 嗎 *ma*, 否 *fǒu* (Td)

As mentioned earlier, some tagsets consider participle forms of verbs, others classify them as adjectives; some tagsets make numerals special cases of adjectives, others have separate POS tags for cardinals, ordinals and various other numeral classes, yet others separate cardinal numbers and put the rest under other POSes. Differences in approaches taken by different tagsets might result in different feature values; for instance, we could decode verbform = “participle” without regard to whether pos = “verb” or pos = “adj”. Naturally it is desirable to decode the same thing into the same set of features each time. Although we could ban particular feature-value combinations in InterSet, effectively forcing the driver authors to seek the permitted decoding, we prefer to leave it as a recommendation, since we do not want to predict, which feature combinations will never ever be needed to distinguish two different words. The recommending guidelines (part of InterSet documentation) are another output of our study.

Probably the broadest source of problems is pronouns, determiners and various WH-words. Somewhere pronouns are only personal or possessive; somewhere there is a diversity of interrogative, relative, demonstrative, indefinite and negative pronouns. In the BulTreeBank (Simov et al. 2004), anything interrogative is a pronoun, although it could be considered numeral (*how much?*) or adverb (*where? when? how?*) elsewhere. Some tagsets address the variable syntactic behavior of pronouns (*I* substitutes a noun, *my* substitutes an adjective). Some tagsets and languages do not have determiners but they have pronouns (demonstrative, indefinite) instead. All that lead us to remove pronouns and determiners as independent parts of speech. Instead, nouns, adjectives and adverbs have the feature “prontype” to distinguish the various types (personal, demonstrative, interrogative...) Empty value of this feature signals a normal noun (adjective, adverb).

Note however, that any guidelines are only to ensure unified approach to different presentations of the same

information. It does not apply to information that simply is not there. If cardinals were tagged as *normal* adjectives (without sub-classing adjectives to numeral and others) they would remain so in InterSet and also in the target tagset. We cannot add information, we only can lose it.

5. Experiments

At the time of writing, 14 drivers have been completed, with quite differing numbers of tags.⁷ Some of the CoNLL tagsets are derived from other tagsets and share their properties (except for Czech, there is a one-to-one mapping between the original and the derived tagset; for Czech, the original PDT tagset is a subset of the CoNLL tagset). Table 1 shows an overview:

Tagset / Driver	Number of tags	Number of tags “other”	Approximate implementation time
ar::conll	241	21	13 h
bg::conll	528	247	35 h
cs::conll	4854	775	6 h
cs::pdt	4288	209	18 h
da::conll	143	6	7 h
de::conll	54	1	10 min
de::stts	54	1	4 h
en::conll	45	2	45 min
en::penn	45	2	3 h
sv::conll	41	12	20 min
sv::hajic	156	17	<i>estimated 8 h</i>
sv::mamba	41	12	3 h
sv::svdahybrid	76	0	<i>estimated 2 h</i>
zh::conll	294	294	21 h

Table 1: Overview of tagset drivers. The “other” column shows tags that make the decoder set the “other” feature.

The working times needed to design particular drivers differ greatly due to various reasons. The Czech tagsets are the most complex but they did not take the most time because the PDT tagset is the native environment for the author. On the other hand, Bulgarian was both complex and differing enough from Czech in approach to pronouns, necessity of introducing new verb tenses, definiteness values etc. Also, the CoNLL conversion of this and other tagsets is quite inconsistent and represents the same feature-value pair in different tags differently. The most exotic tagset w.r.t. this work is the Chinese (Chen and Hsieh 2004) one. Its nearly 300 tags encode mostly things that cannot be represented in InterSet (e.g., there are more than 60 classes of prepositions, containing one to three words each). The intersection of the information encoded by the Chinese tagset with the other tagsets contains only about 10 basic parts of speech. Processing time of Chinese has been further extended because of poor documentation bundled with the CoNLL data.

⁷ For some of the tagsets, the number of tags in the respective corpus has been counted; the true total of possible tags is probably higher.

The processing times are to be compared to time needed to accomplish a targeted conversion for a given tagset pair. Earlier experiments showed us that they are roughly comparable to writing a driver. (We were able to implement conversion from the Russian Dependency Treebank (Boguslavsky et al. 2000) to the Czech PDT tags in about 12 hours; Arabic tags by the Tim Buckwalter’s morphological analyzer (Buckwalter 2002) took about 8 hours. However, drivers presented in Table 1 allow for $14 \times 13 = 182$ conversions, yielding less than 1 hour per conversion on average.

Table 2 illustrates the proportion of information that is shared by tagsets and can be preserved by the conversion. Note that even tagsets for the same language or closely related languages (Danish, Swedish) can be quite divergent due to different corpus designs.

	ar	bg	csc	csp	da	de	en	svh	svm	zh
ar	241	42	68	54	29	17	15	33	12	11
bg	65	528	104	94	64	32	25	50	15	11
csc	68	46	4854	4288	44	21	26	56	14	11
csp	66	42	4288	4288	42	20	24	54	13	11
da	25	46	55	54	143	24	24	71	14	11
de	14	16	17	16	17	54	20	18	15	10
en	16	17	28	26	22	20	45	28	17	11
svh	33	34	63	62	62	22	28	156	17	11
svm	14	15	15	14	15	17	17	16	41	10
zh	10	9	10	10	10	11	9	10	9	294

Table 2: Number of tags resulting from conversion from drivers named in row headers to drivers named in column headers.

As a practical application, driver-based tag conversion has been used in experiments with cross-language parser adaptation from Danish to Swedish (Zeman and Resnik 2008). We have used the reranking parser by (Charniak and Johnson 2005), originally written for the English Penn Treebank. Although the parser can be given a table of symbols from a new corpus, with Interset we could take a much faster approach: we simply converted the Danish and Swedish data to the Penn Treebank format (including the POS tags), and made the parser think it was working with Penn data. Also, converting the divergent Danish and Swedish data to a common tagset was a crucial point in the adaptation technique itself.

Finally, we experimented with a dependency parser that is statistical in nature (Zeman 2004) and can learn dependencies of tags from any tagset; however it contains also many ad-hoc rules that bind it to the format of the Prague Dependency Treebank. The results of the experiments, shown in Table 3, reveal that tagset conversion helps the parser better adapt to new corpora. Experiments have been conducted with the CoNLL data.

Lang	Year	P(orig)	P(conv)	McNemar
ar	2006	64.3	67.6	yes
ar	2007	59.8	66.9	yes
bg	2006	68.0	71.3	yes
cs	2006	56.1	71.4	yes

Lang	Year	P(orig)	P(conv)	McNemar
cs	2007	58.7	74.0	yes
da	2006	68.3	69.8	yes
en	2007	63.8	67.3	yes
sv	2006	71.0	73.5	yes
zh	2006	69.0	68.0	no
zh	2007	66.1	63.5	yes

Table 3: Accuracy of the parser on various CoNLL data sets, using original and converted tags. The last column indicates whether the change was statistically significant, using the McNemar’s test with $p \leq 0.05$.

The decrease of accuracy for Chinese can be easily explained due to the large divergence of the Chinese tagset from the others: too much information gets lost during the conversion.

We are currently experimenting with other parsers (Malt parser (Nivre 2006), MST parser (McDonald et al. 2005)) as well; however, we do not expect significant improvements here, since these parsers are not so heavily dependent on one “home” treebank.

6. Related Work

We do not know about any comparable work applied directly to unified tagset accessing and conversion. There have been however several European projects concerning tagset standardization: EAGLES (EAGLES 1996; Leech and Wilson 1999), LE-PAROLE (Volz and Lenz 1996; Bacelar et al. 1997; etc.), MULTEXT (Ide and Véronis 1994) and MULTEXT-EAST (Erjavec 2004). A common goal of these projects was to create tagging standards and/or multilingual tagged corpora that would share a unified approach. Our work, in contrast, comes up with a method of unifying various tagsets that are “out there” and that need not necessarily conform to the above standards. Various EAGLES-compliant tagsets can be added to our system and their mutual similarity will probably make adding them all easier. We are currently considering making our internal set of features EAGLES-compliant as well.

7. Conclusion

We have proposed a method for tagset conversion that is reusable and, to a reasonable extent, universal. Our interlingua-inspired approach enables to interpret part-of-speech and morphological tags in a uniform way, and to convert information that is shared by two tagsets. Besides the obvious advantage of being able to use tools that expect a particular tagset, we also observed improvements in performance of a statistical parser.

8. Acknowledgements

I would like to thank Philip Resnik for helpful comments and encouragement.

The research reported on in this paper has been supported by Grant No. N00014-01-1-0685 ONR. Ongoing research is supported by the Ministry of Education of the Czech Republic, project

MSM0021620838, and Czech Academy of Sciences, project No. 1ET101470416.

9. References

- Fernanda Bacelar, José Bettencourt, Palmira Marrafa, Ricardo Ribeiro, Rita Veloso, Luzia Wittmann (1997). *LE-PAROLE — Do corpus à modelização da informação lexical num sistema multifunção*. In: Actas do XIII Encontro da Associação Portuguesa de Linguística, Portugal.
- Igor Boguslavsky, Svetlana Grigorieva, Nikolai Grigoriev, Leonid Kreidlin, Nadezhda Frid (2000): *Dependency Treebank for Russian: Concept, Tools, Types of Information*. In: Proceedings of COLING 2000. Saarbrücken, Germany.
- Alena Böhmová, Jan Hajič, Eva Hajičová, Barbora Hladká (2003). *The Prague Dependency Treebank: A Three-Level Annotation Scenario*. In: Anne Abeillé (ed.): *Treebanks: Building and Using Syntactically Annotated Corpora*. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Sabine Buchholz, Erwin Marsi (2006). *CoNLL-X Shared Task on Multilingual Dependency Parsing*. In: Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL). New York, USA.
- Tim Buckwalter (2002). *Buckwalter Arabic Morphological Analyzer Version 1.0*. Linguistic Data Consortium, LDC Catalog No. LDC2002L49, University of Pennsylvania, Philadelphia, USA.
- Eugene Charniak, Mark Johnson (2005). *Coarse-to-Fine N-Best Parsing and MaxEnt Discriminative Reranking*. In: Proceedings of ACL, pp. 173–180. Ann Arbor, Michigan, USA.
- Keh-Jiann Chen, Yu-Ming Hsieh (2004). *Chinese Treebanks and Grammar Extraction*. In: Proceedings of IJCNLP 2004, pp. 560–565. Hainan, China.
- Silvie Cinková, Jan Pomikálek (2006). *LEMPAS: A Make-Do Lemmatizer for the Swedish PAROLE-Corpus*. In: Prague Bulletin of Mathematical Linguistics, vol. 86. Univerzita Karlova, Praha, Czechia.
- EAGLES (1996). Expert Advisory Group on Language Engineering Standards. <http://www.ilc.pi.cnr.it/EAGLES/home.html>
- Tomaž Erjavec (2004). *MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora*. In: Fourth International Conference on Language Resources and Evaluation (LREC 2004). Lisboa, Portugal.
- Jan Hajič, Otakar Smrž, Petr Zemánek, Jan Šnidauf, Emanuel Beška (2004). *Prague Arabic Dependency Treebank: Development in Data and Tools*. In: Proceedings of NEMLAR-2004, pp. 110–117.
- Nancy Ide, Jean Véronis (1994). *MULTEXT (Multilingual Tools and Corpora)*. In: Proceedings of the 15th International Conference on Computational Linguistics (COLING-94). Kyoto, Japan.
- Matthias T. Kromann, Line Mikkelsen, Stine Kern Lyng (2004). *Danish Dependency Treebank*. At <http://www.id.cbs.dk/~mtk/treebank/>. København, Denmark.
- Geoffrey Leech, Andrew Wilson (1999). *Standards for Tagsets*. In: Syntactic Wordclass Tagging. Text, Speech and Language Technology (9), pp. 55–80 Kluwer Academic Publishers, Dordrecht, The Netherlands. ISBN 0792358961.
- Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz (1993). *Building a Large Annotated Corpus of English: the Penn Treebank*. In: Computational Linguistics, vol. 19, pp. 313–330. USA.
- Ryan McDonald, Fernando Pereira, Kiril Ribarov, Jan Hajič (2005). *Non-projective Dependency Parsing Using Spanning Tree Algorithms*. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 523–530. Vancouver, Canada.
- Joakim Nivre (2006). *Inductive Dependency Parsing*. Dordrecht: Springer (Text, speech and language technology series ed. by Nancy Ide and Jean Véronis, vol. 34), xi+216 pp., ISBN 1-4020-4888-2.
- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, Deniz Yuret (2007). *The CoNLL 2007 Shared Task on Dependency Parsing*. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 915–932. Praha, Czechia.
- Joakim Nivre, Jens Nilsson, Johan Hall (2006). *Talbanken05: A Swedish Treebank with Phrase Structure and Dependency Annotation*. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006). May 24–26. Genova, Italy.
- Richard Hook Richens (1958). *Interlingual Machine Translation*. In: The Computer Journal 1958 1(3):144–147. British Computer Society, United Kingdom.
- Kiril Simov, Petya Osenova, Milena Slavcheva (2004). *BTB-TR03: BulTreeBank Morphosyntactic Tagset*. BulTreeBank Project Technical Report No. 03. Sofija, Bulgaria.
- Christine Stöckert, Christine Thielen, Anne Schiller, Simone Teufel (1995). *Stuttgart Tübingen Tagset*. Vorläufige Guidelines für das Tagging deutscher Textcorpora mit STTS. At: <http://www.sfs.nphil.uni-tuebingen.de/Elwis/stts/stts.html>
- Norbert Volz, Suzanne Lenz (1996). *Multilingual Corpus Tagset Specifications*. MLAP PAROLE 63–386 WP 4.1.4. IDS, Mannheim, Germany.
- Daniel Zeman (2004). *Parsing with a Statistical Dependency Model (Ph.D. thesis)*. Univerzita Karlova, Praha, Czechia.
- Daniel Zeman, Philip Resnik (2008). *Cross-Language Parser Adaptation between Related Languages*. In: Proceedings of IJCNLP workshop on NLP of less privileged languages (NLPLPL). Hyderabad, India.

6.3 HamleDT: Harmonized Multi-language Dependency Treebank

Full reference: Daniel Zeman, Ondřej Dušek, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zdeněk Žabokrtský, and Jan Hajič. HamleDT: Harmonized multi-language dependency treebank. *Language Resources and Evaluation*, 48:601–637, 2014. URL <https://link.springer.com/content/pdf/10.1007/s10579-014-9275-2.pdf>. [Zeman et al., 2014]

Comments: The first paper about HamleDT (Section 3.1) was Zeman et al. [2012], presented at LREC in İstanbul. This is an extended version of that paper, which we were invited to submit to the LRE journal. HamleDT was a pioneering project, which provided the first collection of harmonized treebanks; it was also the largest one. Later at LREC in Reykjavík we presented a new version of HamleDT, which provided an alternative conversion of the treebanks to Stanford Dependencies [Rosa et al., 2014]. When the Universal Dependencies initiative started in 2014, the consensus was reached that the syntactic annotation in UD will be derived from Stanford (rather than Prague) dependencies. During 2015, we made all HamleDT treebanks compatible with the new UD standard. We made one final release, HamleDT 3.0. All HamleDT treebanks with permissive licenses were then incorporated in UD, which became a successor of HamleDT. My contribution: about 25%. Number of citations according to Google Scholar (retrieved 2023-07-21): **84**, together with the other two papers: **215**.

HamleDT: Harmonized multi-language dependency treebank

Daniel Zeman · Ondřej Dušek · David Mareček · Martin Popel · Loganathan Ramasamy · Jan Štěpánek · Zdeněk Žabokrtský · Jan Hajič

Published online: 26 August 2014
© Springer Science+Business Media Dordrecht 2014

Abstract We present HamleDT—a *HARmonized Multi-LanguagE Dependency Treebank*. HamleDT is a compilation of existing dependency treebanks (or dependency conversions of other treebanks), transformed so that they all conform to the same annotation style. In the present article, we provide a thorough investigation and discussion of a number of phenomena that are comparable across languages, though their annotation in treebanks often differs. We claim that transformation procedures can be designed to automatically identify most such phenomena and convert them to a unified annotation style. This unification is beneficial both to comparative corpus linguistics and to machine learning of syntactic parsing.

Keywords Dependency treebank · Annotation scheme · Harmonization

1 Introduction

Growing interest in dependency parsing is accompanied (and inspired) by the availability of new treebanks for various languages. Shared tasks such as CoNLL 2006–2009 (Buchholz and Marsi 2006; Nivre et al. 2007; Surdeanu et al. 2008; Hajič et al. 2009) have promoted parser evaluation in multilingual settings. However, differences in parsing accuracy in different languages cannot be always attributed to language differences. They are often caused by variation in domains, sizes and annotation styles of the treebanks. The impact of data size can be estimated by learning curve experiments, but normalizing the annotation style is difficult. We present a method to transform the treebanks into a common style, including a software that implements the method. We have studied treebanks of 29

D. Zeman (✉) · O. Dušek · D. Mareček · M. Popel · L. Ramasamy · J. Štěpánek · Z. Žabokrtský · J. Hajič
Faculty of Mathematics and Physics, ÚFAL, Charles University in Prague, Prague, Czech Republic
e-mail: zeman@ufal.mff.cuni.cz

languages and collected a long list of variations.¹ We propose one common style (called HamleDT v1.5 style) and provide a transformation from original annotations to this style for almost all² the phenomena we identified. In addition to dependency tree structure normalization, we also unify the tagsets of both the part-of-speech/morphological tags and the dependency relation tags.

The motivation for harmonizing the annotation conventions used for different treebanks was already described in literature, e.g., by McDonald et al. (2013). Clearly, a unified representation of language data is supposed to facilitate the development of multilingual technologies. The harmonized set of treebanks should improve the interpretability and comparability of parsing accuracy results, and thus help to drive the development of dependency parsers towards multilingual robustness. For instance, the range of unlabeled attachment scores reached by a typical state-of-the-art supervised dependency parser in different languages spans an interval of around 10 % points (given training data of a comparable size) and is even bigger for unsupervised parsers, as documented, e.g., by Mareček and Žabokrtský (2012). It is not entirely clear whether and to what extent this variance can be attributed to the peculiarities of the individual languages, or merely to the choice of annotation conventions used for the language. Using HamleDT should make it possible to separate these two sources of variance. Besides supervised and unsupervised multilingual parsing, homogeneity of the data is also essential for experiments on cross-lingual transfer of syntactic structures, be it based on projecting trees (Hwa et al. 2005) or on transferring delexicalized models (McDonald et al. 2011a).

The common style defined in HamleDT v1.5 serves as a reference point: the ability to say “our results are based on HamleDT v1.5 transformations of treebank XY” will facilitate the comparability of future results published in all these subfields.

The purpose of HamleDT is not to find a single choice of annotation conventions that ideally suits all possible tasks concerning syntactic structures, as this is hardly to be expected doable. However, assuming a different annotation convention fits a particular task better, it is much simpler to transform all the treebanks to the desired shape after they have been collected and unified in HamleDT.

Last but not least, we believe that the unified representation of linguistic content may be advantageous for linguists, enabling them to compare languages based on treebank material without the need to study multiple annotation guidelines.

2 Related work

There have been a few attempts recently to address the same problem, namely:

- Schwartz et al. (2012) define two measures of syntactic *learnability* and evaluate them using five different parsers on varying annotation styles of six phenomena

¹ The initial version has been described in Zeman et al. (2012).

² HamleDT v1.5 does not include the harmonization of verbal groups (see Sect. 5.4).

(coordination, infinitives, noun phrases, noun sequences, prepositional phrases and verb groups). They work only with English; they generate varying annotations during the conversion of the Penn TreeBank WSJ corpus (Marcus et al. 1993) constituency annotation to dependencies.

- Tsarfaty et al. (2011) compare the performance of two parsers on different constituency-to-dependency conversions of the (English) Penn Treebank. They do not see the solution in data transformations; instead, they develop an evaluation technique that is robust with respect to some³ annotation styles.
- McDonald et al. (2011b) experiment with cross-language parser training, relying on a rather small universal set of part-of-speech tags. They do not transform syntactic structures, however. They note that different annotation schemes across treebanks are responsible for the fact that some language pairs work better together than others. They use English as the source language and Danish, Dutch, German, Greek, Italian, Portuguese, Spanish, and Swedish as target languages.
- Seginer (2007) discusses possible annotation schemes for coordination structures and relative clauses in relation to his *common cover link* representation.
- Bosco et al. (2010) compare three different dependency parsers developed and tested with respect to two Italian treebanks.
- Bengoetxea and Gojenola (2009) evaluate three types of transformations on Basque: transformation of subordinate sentences, coordinations and projectivization. An important difference between their approach and ours is that their transformations can change tokenization.
- Nilsson et al. (2006) show that transformations of coordination and verb groups improve parsing of Czech.

3 Data

We identified over 30 languages for which treebanks exist and are available for research purposes. Most of them can either be acquired free of charge or are included in the Linguistic Data Consortium⁴ membership fee.

Most of the treebanks are natively based on dependencies, but some were originally based on constituents and transformed via a head-selection procedure. For instance, Spanish phrase-structure trees were converted to dependencies using the method of Civit et al. (2006).

HamleDT v1.5 currently covers 29 treebanks, with several others to be added soon. Table 1 lists the treebanks along with their data sizes. In the following, we use ISO 639 language codes in square brackets to refer to the treebanks of these languages, so e.g. [en] refers to the English treebank. A list of all 29 treebanks with references is included in Appendix 1.

³ The transformations are not robust to coordination styles.

⁴ <http://www ldc.upenn.edu/>.

Table 1 Overview of data resources included in HamleDT v1.5

Language	Prim. tree type	Used data source	Sents.	Tokens	Train/test (% sents)	Avg. sent. length	Nonprj. deps. (%)
Arabic (ar)	dep	C2007	3,043	116,793	96/4	38.38	0.37
Basque (eu)	dep	prim	11,226	151,604	90/10	13.50	1.27
Bengali (bn)	dep	I2010	1,129	7,252	87/13	6.42	1.08
Bulgarian (bg)	phr	C2006	13,221	196,151	97/3	14.84	0.38
Catalan (ca)	phr	C2009	14,924	443,317	88/12	29.70	0.00
Czech (cs)	dep	C2007	25,650	437,020	99/1	17.04	1.91
Danish (da)	dep	C2006	5,512	100,238	94/6	18.19	0.99
Dutch (nl)	phr	C2006	13,735	200,654	97/3	14.61	5.41
English (en)	phr	C2007	18,577	446,573	99/1	24.03	0.33
Estonian (et)	phr	prim	1,315	9,491	90/10	7.22	0.07
Finnish (fi)	dep	prim	4,307	58,576	90/10	13.60	0.51
German (de)	phr	C2009	38,020	680,710	95/5	17.90	2.33
Greek (el)	dep	C2007	2,902	70,223	93/7	24.20	1.17
Greek (grc)	dep	prim	21,160	308,882	98/2	14.60	19.58
Hindi (hi)	dep	I2010	3,515	77,068	85/15	21.93	1.12
Hungarian (hu)	phr	C2007	6,424	139,143	94/6	21.66	2.90
Italian (it)	dep	C2007	3,359	76,295	93/7	22.71	0.46
Japanese (ja)	dep	C2006	17,753	157,172	96/4	8.85	1.10
Latin (la)	dep	prim	3,473	53,143	91/9	15.30	7.61
Persian (fa)	dep	prim	12,455	189,572	97/3	15.22	1.77
Portuguese (pt)	phr	C2006	9,359	212,545	97/3	22.71	1.31
Romanian (ro)	dep	prim	4,042	36,150	93/7	8.94	0.00
Russian (ru)	dep	prim	34,895	497,465	99/1	14.26	0.83
Slovene (sl)	dep	C2006	1,936	35,140	79/21	18.15	1.92

Table 1 continued

Language	Prim. tree type	Used data source	Sents.	Tokens	Train/test (% sents)	Avg. sent. length	Nonprj. deps. (%)
Spanish (es)	phr	C2009	15,984	477,810	90/10	29.89	0.00
Swedish (sv)	phr	C2006	11,431	197,123	97/3	17.24	0.98
Tamil (ta)	dep	prim	600	95,81	80/20	15.97	0.16
Telugu (te)	dep	I2010	1,450	5,722	90/10	3.95	0.23
Turkish (tr)	dep	C2007	5,935	69,695	95/5	11.74	5.33

The average sentence length is the number of tokens divided by the number of sentences. Varying tokenization schemes obviously influence the numbers; see Sect. 5.7 for details on the individual languages. The *C* code in the fourth column means “CoNLL shared task”, *I* means “ICON” and *prim* means primary (non-shared-task) source. The last column gives the percentage of nodes attached non-projectively

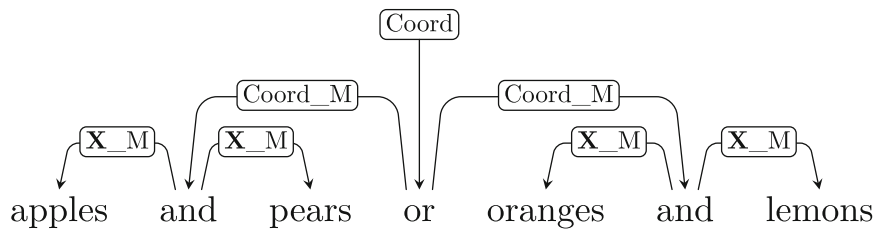



Fig. 1  Nested coordination in the Prague style. X represents the relation of the whole structure to its parent. *_M* denotes *members* of coordination, i.e., conjuncts

Many treebanks (especially those used in CoNLL shared tasks) define a train/test data split. This is important for the comparability of experiments with automated parsing and part-of-speech tagging. We preserve the original data division and define test subsets for the remaining treebanks as well. In doing so, we try to keep the test size similar to the majority of CoNLL 2006/2007 test sets, i.e., roughly 5,000 tokens.

Throughout this article, a *dependency tree* is an abstract structure of *nodes* and *dependencies* that capture syntactic relations in a sentence. Nodes correspond to the *tokens* of the sentence, i.e. to words, numbers, punctuation and other symbols (see Sect. 5.7 for more on tokenization). Besides the actual word form, the node typically holds additional attributes of the token, such as its lemma and part of speech. Dependencies are directed arcs between nodes. Every node *is attached to* (*depends on*) exactly one other node, called its *parent*. We draw the dependency as an arrow going from the parent to the *child*. Thus every node has one incoming dependency and any number of outgoing dependencies. There is one exception: an artificial *root node* that does not correspond to any real token and has only outgoing dependencies. Dependencies have *labels* that mark the type of the relation.

Most diagrams in this article (Fig. 1 and onwards) depict just a snippet of the sentence, i.e. a *subtree*. Selected tokens (word forms) are shown in a sequence respecting the word order, with dependencies drawn as labeled arrows between two tokens (nodes). The artificial root of the whole sentence is never shown; the root token of the subtree has one incoming dependency going straight down (from an invisible parent). The relation between the subtree and its invisible parent is labeled X (it does not make sense to show the real relation type without the parent).

4 Harmonization

Our effort aims at identifying all syntactic constructions that are annotated differently in different treebanks. Once a particular construction is identified, we can typically find all its instances in the treebank using existing syntactic and morphological tags, i.e., with little or no lexical knowledge. Thanks to this fact, we were able to design algorithms to normalize the annotations of many linguistic phenomena to a single style, which we refer to as the HamleDT v1.5 style.

The HamleDT v1.5 style is mostly derived from the annotation style of the Prague Dependency Treebank (PDT, Hajič et al. 2006).⁵ This is a matter of convenience, to a large extent: This is the scheme with which the authors feel most at home, and many of the included treebanks already use a style similar to PDT. We do not want to claim that the HamleDT v1.5 style is objectively better than other styles. (Please note, however, that in case of coordination, the HamleDT v1.5 style provides a more expressive power than the other options, as described in Sect. 5.1).

The normalization procedure involves both structural transformations and changes to dependency relation labels. While we strive to design the structural transformations to be as reversible as possible, we do not attempt to save all information stored in the dependency labels. The original⁶ labels vary widely across treebanks, ranging from very simple, e.g., NMOD “generic noun modifier” in [en], over standard *subject*, *object*, etc. relations, to deep-level functions of Pāṇinian grammar such as *karta* and *karma* (k1 and k2) in [hi, bn, te].⁷ It does not seem possible to unify these tagsets without relabeling whole treebanks manually.

We use a lossy scheme that maps the dependency labels on the moderately sized tagset of PDT analytical functions⁸—see Table 2.

Occasionally the original structure and dependency labels are not enough to determine the normalized output. For instance, the German label RC is assigned to all dependencies that attach a subordinate clause to its parent. The set of HamleDT v1.5 labels distinguishes clauses that act as nominal attributes (A_{tr}) from those that substitute adverbial modifiers (A_{dv}). We look at the part of speech of the parent: if it is a noun, we label the dependency A_{tr}; if it is a verb, we label it A_{dv}.⁹ Thus we also consider the part of speech, the word form, or even further morphological properties. Since the morphological (part-of-speech) tagsets also vary greatly across treebanks, we use the Interset approach described by Zeman (2008) to access all morphological information. Interset is a kind of interlingua for parts of speech and morphosyntactic features. Its aim is to provide a unified representation for as many feature values in existing tagsets as possible. We created converters (“drivers”) to

⁵ So far, there are only two differences between the PDT style (used in [cs]) and the HamleDT v1.5 style: handling of appositions (see Table 3) and marking of conjuncts (in HamleDT, the root of a conjunct subtree is marked as conjunct even if it is a preposition or subordinating conjunction; in PDT, only content words are marked as conjuncts). By conjunct, we mean a member of coordination (unlike Quirk et al. 1985). By content word, we mean autosemantic word, i.e. a word with a full lexical meaning, as contrasted with auxiliary. Note that PDT also has a more abstract layer of annotation (called *tectogrammatical*), but in this work, we only use the shallow dependencies (called *analytical* layer in PDT).

⁶ Unless we explicitly say otherwise, we mean by “original” the data source indicated in Table 1. It may actually differ from the *really original* treebank. For instance, some of the CoNLL data underwent a conversion procedure to the CoNLL format from other formats, and some information may have been lost in the process.

⁷ In the Pāṇinian tradition, *karta* is the agent, doer of the action, and *karma* is the “deed” or patient. See Bharati et al. (1994).

⁸ They are approximately the same as the dependency relation labels in the Czech CoNLL data set. To illustrate the mapping, more details on [bn] and [en] conversion are presented in Tables 4 and 5 in Appendix 2.

⁹ Ideally we would also want to distinguish objects (O_{bj}) from adverbials. Unfortunately, this particular source annotation does not provide enough information to make such a distinction.

Table 2 Selected types of dependency relations and their relative frequency in the harmonized treebanks

Language	Atr	Adv	Obj	AuxP	Sb	Pred	Coord	AuxV	AuxC	Rest
Arabic (ar)	36.5	6.4	9.1	14.2	6.3	3.1	4.0	0.0	2.3	18.2
Basque (eu)	19.6	24.0	8.7	0.0	7.2	5.7	3.4	8.3	1.0	22.2
Bengali (bn)	18.2	22.7	17.9	0.0	16.6	16.7	4.9	0.0	0.0	3.0
Bulgarian (bg)	23.3	8.8	12.8	14.6	7.7	7.3	3.1	0.8	3.3	18.4
Catalan (ca)	22.4	16.7	5.2	9.9	7.4	8.1	2.9	9.3	1.8	16.4
Czech (cs)	28.5	10.4	8.1	9.9	7.1	6.0	4.1	1.2	1.7	23.1
Danish (da)	23.8	12.2	12.1	10.7	9.8	5.3	3.4	0.0	3.4	19.3
Dutch (nl)	14.1	24.7	6.8	10.3	8.5	7.4	2.1	5.2	3.7	17.2
English (en)	30.0	12.0	5.7	9.8	7.9	4.3	2.2	4.0	1.8	22.2
Estonian (et)	12.8	25.7	6.6	5.9	13.0	14.1	1.3	2.6	0.6	17.4
Finnish (fi)	29.7	18.2	7.8	1.5	9.4	8.3	4.1	1.6	1.2	18.2
German (de)	31.2	11.8	10.4	10.1	7.9	5.3	2.8	0.5	1.2	18.7
Greek (grc)	15.4	13.0	14.2	3.8	7.7	8.6	6.5	0.0	1.4	29.4
Greek (el)	39.8	9.9	7.5	8.3	7.1	4.5	3.2	4.0	1.6	14.0
Hindi (hi)	26.8	13.4	9.6	21.1	6.8	5.3	2.4	6.3	1.6	6.8
Hungarian (hu)	30.4	13.9	5.2	1.6	5.9	8.3	2.4	1.3	1.6	29.2
Italian (it)	22.2	12.4	4.9	14.7	5.2	4.8	3.3	2.8	1.1	28.5
Japanese (ja)	11.5	16.6	0.6	5.8	3.4	7.3	0.3	0.0	0.0	54.6
Latin (la)	17.9	13.7	15.9	5.3	10.6	8.8	6.6	1.1	3.1	17.2
Persian (fa)	25.3	8.8	10.0	14.0	6.4	7.7	4.1	0.1	2.7	20.8
Portuguese (pt)	24.6	24.0	7.1	11.4	6.0	4.3	2.4	0.0	1.0	19.0
Romanian (ro)	27.7	13.3	7.2	17.6	8.5	11.2	1.8	7.7	0.0	5.0
Russian (ru)	30.4	16.9	16.3	12.3	10.4	6.2	4.0	0.0	1.6	1.9
Slovene (sl)	15.0	10.9	8.1	7.3	5.9	7.2	4.3	9.4	3.7	28.1
Spanish (es)	22.8	16.9	5.1	9.0	7.8	8.7	2.8	8.0	2.0	17.0
Swedish (sv)	19.3	19.5	6.9	9.3	10.8	6.4	3.9	2.5	2.7	18.8
Tamil (ta)	27.7	0.0	9.7	3.0	7.3	6.0	1.6	6.3	2.8	35.6
Telugu (te)	7.3	21.3	19.5	0.0	19.2	25.6	3.5	0.1	0.0	3.6
Turkish (tr)	38.5	8.0	10.8	1.9	6.9	9.5	3.8	0.0	1.4	19.2
Average	26.2	13.9	8.9	10.3	7.6	6.3	3.3	2.8	1.8	18.8

One can see repeated patterns in the table such as the dominance of adverbials and attributes, or the relatively stable proportion of subjects. However, the numbers are still biased by imperfections in the conversion procedures (e.g., unrecognized AuxV in certain languages)

Atr Attribute, *Adv* adverbial, *Obj* object, *AuxP* preposition, *Sb* subject, *Pred* predicate, *Coord* coordinating conjunction, *AuxV* auxiliary verb, *AuxC* subordinating conjunction

Intersect from all treebank tagsets for which it had not already been available. The normalized treebanks thus provide Intersect-unified morphology as well.

In a typical scenario, the harmonization steps are ordered as follows:

1. file format conversion (from various proprietary formats to a common-schema XML) and character encoding conversion (to UTF-8),

2. conversion of morphological tags to the Intersect tagset,
3. conversion of dependency relation labels to the set of HamleDT labels,
4. conversion of coordination structures into the HamleDT style (i.e., distinguishing members of coordination and shared modifiers, and attaching them to the main coordination conjunction),
5. other changes in the tree structure (i.e., rehangng nodes to make the dependent-governor relations comply with the HamleDT conventions, including relation orientation) and possibly further refinements of the dependency labels.

The last two points (tree transformations) represent the main focus of the present study; many detailed examples are provided in Sect. 5.

The implementation of file format converters is relatively straightforward, even though reverse engineering is sometimes needed due to missing technical documentation.

When implementing the Intersect converters, around 200–500 lines of Perl code are typically needed; the code is usually not very challenging from the algorithmic point of view, but requires a very good insight into the annotation guidelines of the respective resource.

Mapping of dependency labels is usually relatively simple to implement too: sometimes it is enough just to recode the original label (e.g. *Subj* to *Sb*), sometimes the decision must be conditioned by the POS value of the node or of its parent, sometimes the rules are conditioned lexically or by certain structural properties of the tree. However, it all can be done relatively reliably.

More or less the same holds for rehangng the nodes in the fifth step. Typically, there are just a few dozens of transformation rules needed for the third and fifth step (i.e., around 200 lines of Perl code).

The algorithmically most complex step in the harmonization is typically a proper treatment of coordination structures because resolving a coordination structure affects at least three nodes in most cases, coordinations can be nested, and they can combine with almost any dependency relation type. In addition, there are multiple different encodings of coordination structures used in treebanks (17 in HamleDT v1.5), as analyzed in depth by Popel et al. (2013).

Performing the normalization of coordination structures before the normalization of other relations brings about an important advantage: in step 5, it is possible to work with dependent-governor pairs of nodes in the sense of dependency (not just with child–parent node pairs as stored in the trees), disregarding whether the former or the latter (or both) are coordinated. Without this abstraction, even simple operations, such as swapping the relation orientation between nouns and prepositions, would become quite cumbersome, as one would have to keep all possible combinations in mind, e.g. “with A and B”, “with A and with B”, “with A and B or with C and D”, “with or without A”, etc. For more details, please refer to concrete examples in Sect. 5.

5 Annotation styles for various phenomena

In this section, we present a selection of phenomena that we observed and, to various degrees for various languages, included in our normalization scenario.

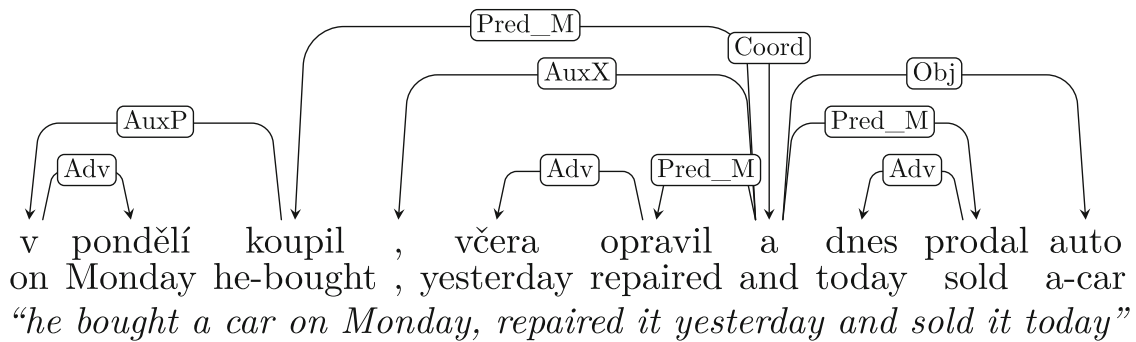
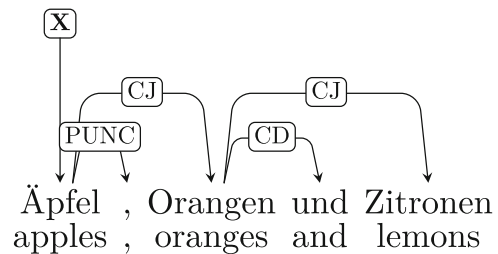


Fig. 2 Shared and private modifiers in the Prague style [cs]: *Car (auto)* is an object shared by all three verbs while the adverbials (*on Monday, yesterday, today*) are private. The whole structure is in the predicate relation to its parent (which is probably the sentence root), so using the notation of Fig. 1: X = Pred

Fig. 3 Coordination in the Mel'čukian style as seen in [de]



Language codes in brackets give examples of treebanks where the particular approach is employed. The symbol in figure captions marks artificial examples. Figures not marked with contain genuine examples found in real data, though some of them have been shortened.

Dependency relation labels from the original treebanks that appear in figures are briefly explained in Appendix 3.

5.1 Coordination

Capturing coordination in a dependency framework has been repeatedly described as difficult for both treebank designers and parsers (and it is generally regarded as an inherent difficulty of dependency syntax as such). Our analysis revealed four families of approaches, which may further vary in the attachment of punctuation, shared modifiers, etc.:

- *Prague* (Figs. 1, 2, 8). All conjuncts are headed by the conjunction. Used in [ar, bn, cs, el, en, eu, grc, hi, la, nl, sl, ta, te] (Hajič et al. 2006).
- *Mel'čukian* (Fig. 3). The first/last conjunct is the head, others are organized in a chain. Used in [de, ja, ru, sv, tr] (Mel'čuk 1988).
- *Stanford* (Fig. 4). The first/last conjunct is the head, others are attached directly to it. Used in [bg, ca, es, fi, it, pt] (de Marneffe and Manning 2008). And
- *Tesnièrean* (Fig. 5). There is no common head, all conjuncts are attached directly to the node modified by the coordination structure. Used in [hu] (Tesnière 1959).

Fig. 4 Coordination in the Stanford style as seen in [ca]

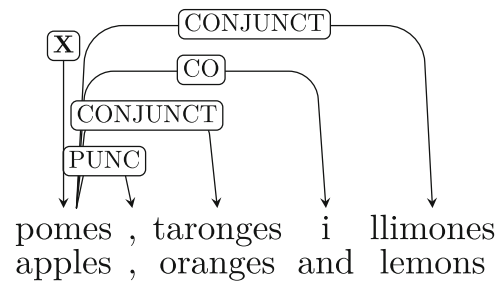


Fig. 5 Coordination in the Tesnièreian style as seen in [hu]. All participating nodes are attached directly to the parent of the coordination

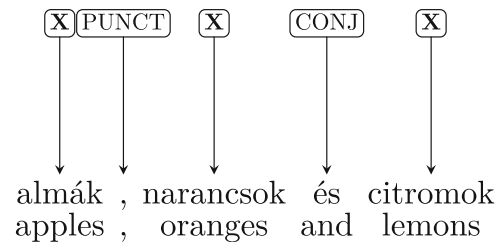
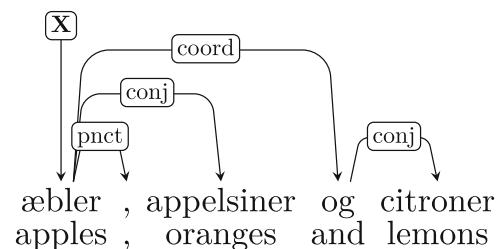


Fig. 6 Danish mixture of Stanford and Mel'čukian coordination styles



Furthermore, the Prague style provides for nested coordinations, as in *apples and pears or oranges and lemons* (see Fig. 1). The asymmetric treatment of conjuncts in the other styles makes nested coordination difficult to read or even impossible to capture in some situations. The Prague style also distinguishes between shared modifiers, such as the subject in *Mary came and cried*, from private modifiers of the conjuncts, as in *John came and Mary cried* (see Fig. 2). Because this distinction is missing in non-Prague-style treebanks, we cannot recover it reliably. We apply several heuristics, but in most cases, the modifiers of the head conjunct are classified as private modifiers.

Danish (Fig. 6) employs a mixture of the Stanford and Mel'čukian styles where the last conjunct is attached indirectly via the conjunction. The Romanian and Russian treebanks omit punctuation tokens (they do not have corresponding nodes in the trees); in the case of Romanian, this means that coordinations of more than two conjuncts are disconnected (Fig. 7).

Given the advantages described above, we decided to use the Prague style (in its [cs] flavor) in our harmonized data. There is just one drawback that we are aware of: Occasionally, there may be no node suitable for the coordination head. Most asyndetic constructions do not pose a problem because there are commas or other punctuation. Without punctuation, the Prague style would need an extra node—that solution has been adopted by the authors of the [ta] treebank (see Fig. 8). Note that

Fig. 7 [ro] uses Prague coordination style mixed with Tesnièrean because punctuation is missing from data

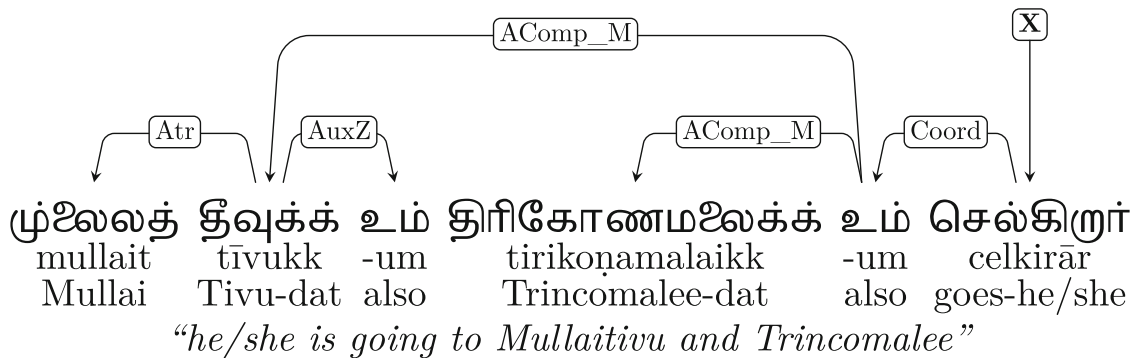
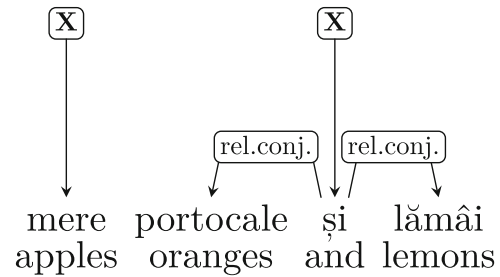



Fig. 8 Coordination in [ta]: The coordinating function is performed by the two morphological suffixes *-um*. They had to be made separate nodes during tokenization because [ta] uses the Prague style and no other coordination head was available except these morphological indicators

one-half of our treebanks already use the Prague style as their native approach, thus they always have a coordination head. In the other half, a fraction of coordinate structures cannot be fully converted (unless we add a new node, which we do not in the current version of HamleDT). For example, 14 out of the 5,988 coordinate structures in [bg] (0.23 %) lack any conjunction or punctuation that could be made the head. In these cases we currently use the first conjunct instead, effectively backing off to the Stanford style.

5.2 Prepositions

Prepositions (or postpositions; Figs. 9, 10, 11) can either govern their noun phrase (NP) [cs, en, sl, ...] or modify the head of its NP [hi]. When they govern the NP, other modifiers of the main noun are attached either to the noun (in most cases) or to the preposition [de]. The label of the relation of the prepositional phrase to its parent is sometimes found at the preposition [de, en, nl]. Elsewhere, the preposition gets an auxiliary label (such as *AuxP* in PDT) despite serving as head, and the real label is found at the NP head [cs, sl, ar, el, la, grc].

In HamleDT v1.5 style, prepositions govern their noun phrase because 1. they may govern the form of the noun phrase (e.g. [cs, ru, sl, de]) and 2. this is the approach taken in most of the treebanks we studied. Other modifiers inside the prepositional phrase, such as determiners and adjectives, should depend on the embedded noun phrase. The preposition is labeled with the auxiliary tag *AuxP* and the real relation between the prepositional phrase and its parent is labeled at the NP head.

Fig. 9  A prepositional phrase in [cs]

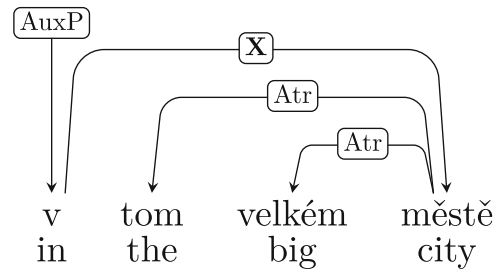



Fig. 10  A prepositional phrase in [de]

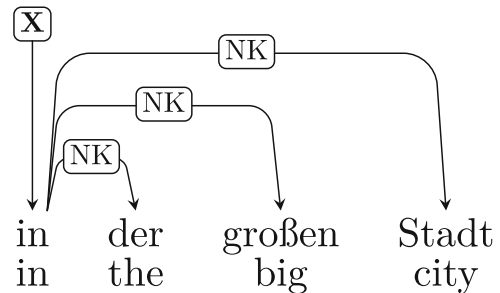

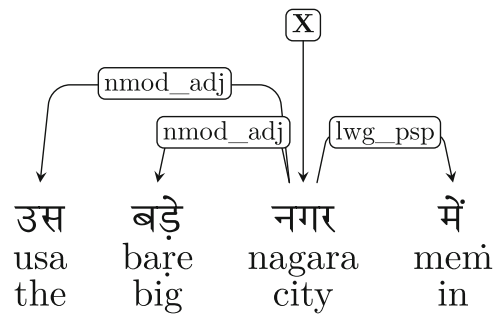


Fig. 11  A postpositional phrase in [hi]



5.3 Subordinate clauses

There are three main types of subordinate clauses:

- *Relative clauses* They modify noun phrases. Typically they are marked by relative pronouns that represent the modified noun and its function within the relative clause. Example: *The man who came yesterday.*
- *Complement clauses* They serve as arguments of predicates, typically verbs. They are marked by subordinating conjunctions. Example: *The man said that he came yesterday.*
- *Adverbial clauses* They modify predicates in the same way as adverbs; but they are not selected as arguments. Example: *If the man comes today he will say more.*

Roots (predicates) of relative clauses are usually attached to the noun they modify, e.g., in *the man who came yesterday*, *came* would be attached to *man* and *who* would be attached to *came* as its subject.

The predicate-modifying clauses use a subordinating conjunction (complementizer, adverbializer) to express their relation to the governing predicate. In

treebanks, the conjunction is either attached to the predicate of the subordinate clause [es, ca, pt, de, ro] (Fig. 12) or it lies between the embedded clause and the main predicate it modifies [cs, en, hi, it, la, ru, sl] (Fig. 13). In the latter case, the label of the relation of the subordinate clause to its parent can be assigned to the conjunction [en, hi, it] or to the clausal predicate [cs, la, sl] (Fig. 14). The comma before the conjunction is attached either to the conjunction or to the subordinate predicate.

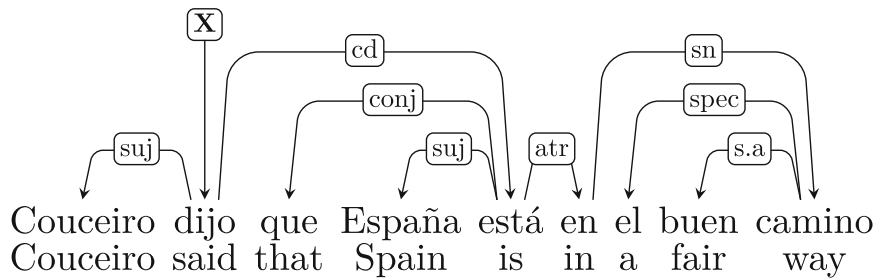


Fig. 12 Subordinate clause in [es]

Fig. 13 Subordinate clause in [it]

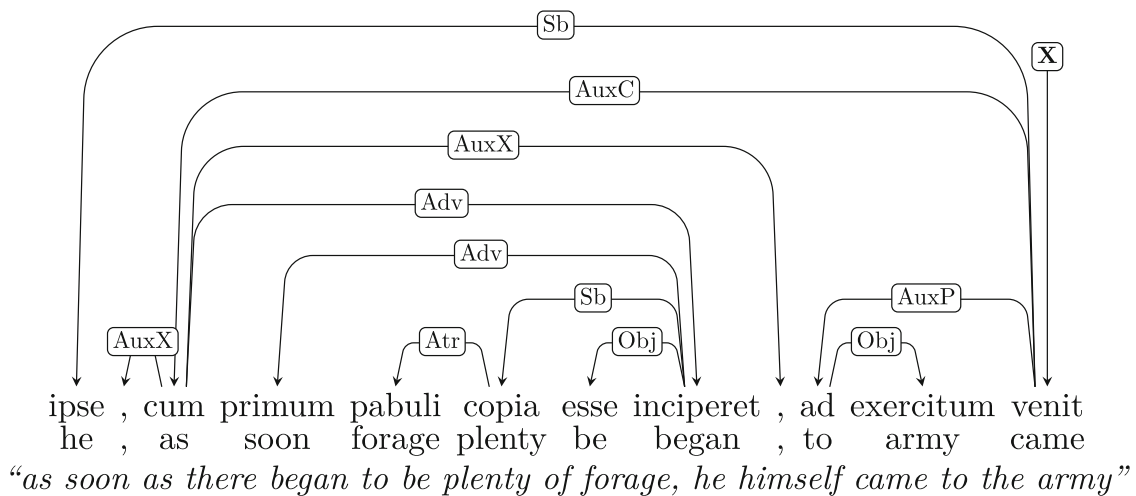
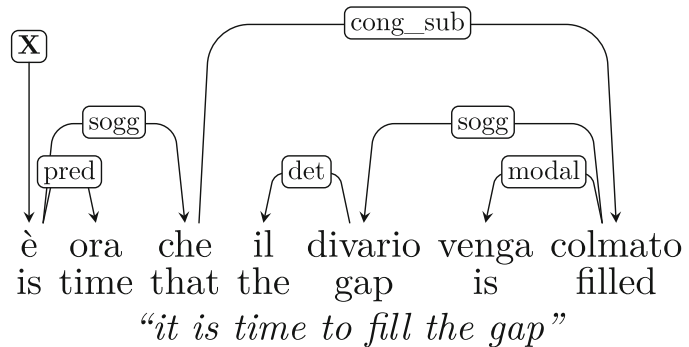


Fig. 14 Subordinate clause in [la]

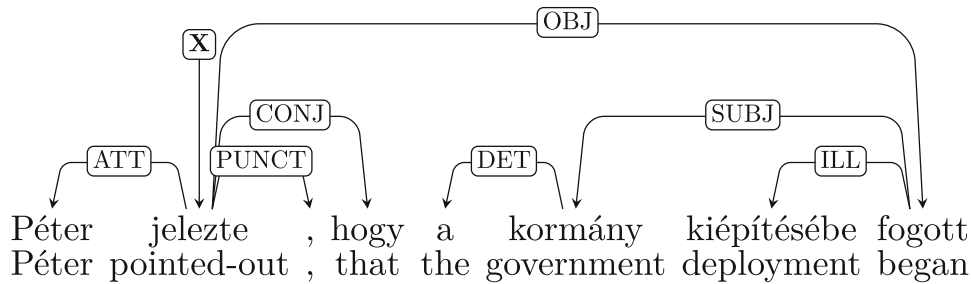


Fig. 15 Subordinate clause in [hu]

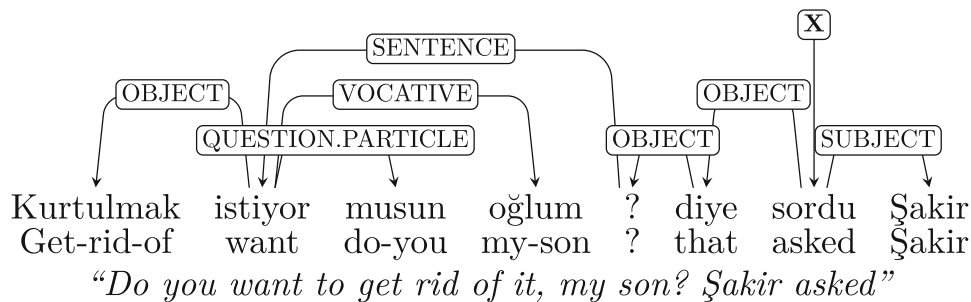


Fig. 16 Subordinate clause (direct speech) in [tr]: Here, the question mark serves as the head of the direct question

The subordinating conjunction may also be attached as a sibling of the subordinate clause [hu], an analogy to the Tesnièrean coordination style (Fig. 15). In Fig. 16, a direct question in [tr] is rooted by the question mark, which is attached to a subordinating postposition.

The Romanian treebank is segmented into clauses instead of sentences, so every clause has its own tree, and inter-clausal relations are not annotated.

HamleDT v1.5 style follows the [cs, sl, la] approach to subordinate clauses (see Figs. 14, 19).

5.4 Verb groups

Various sorts of verbal groups include analytical verb forms (such as auxiliary + participle), modal verbs with infinitives, and similar constructions. Dependency relations, both internal (between group elements) and external (leading to the parent on the one side and verb modifiers on the other side), may be defined according to various criteria: content verb versus auxiliary, finite form versus infinitive, or subject-verb agreement, which typically holds for finite verbs, sometimes for participles but not for infinitives (Figs. 17, 18).

Participles often govern auxiliaries [es, ca, it, ro, sl] (Figs. 19, 20); elsewhere the finite verb is the head [pt, de, nl, en, sv, ru] (Figs. 17, 22, 23), and finally, [cs] mixes both approaches based on semantic criteria. In [hi, ta], the content verb, which could be a participle or a bare verb stem, is the head, and auxiliaries (finite or participles) are attached to it (Figs. 21, 24, 25, 26, 27).

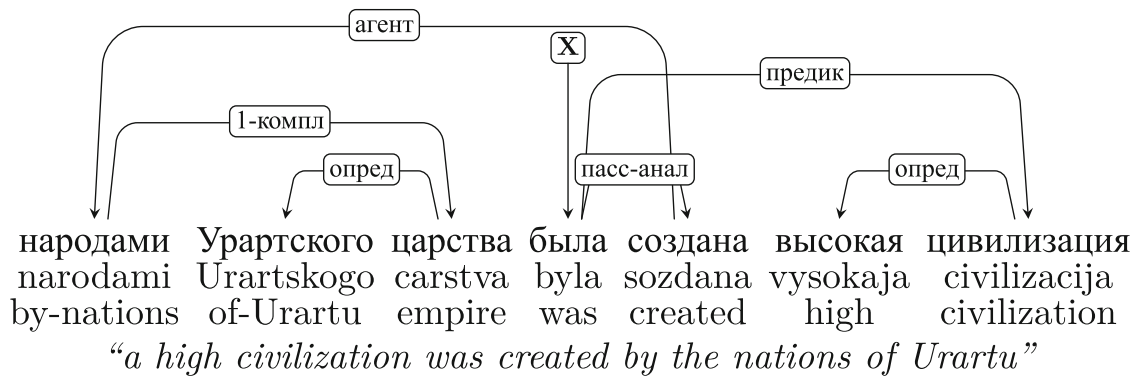


Fig. 17 Passive construction in [ru]: Finite auxiliary verb (*была*) is the head, passive participle (*создана*) depends on it. As a result, the agent (*народами*) is attached non-projectively to the participle (*создана*)

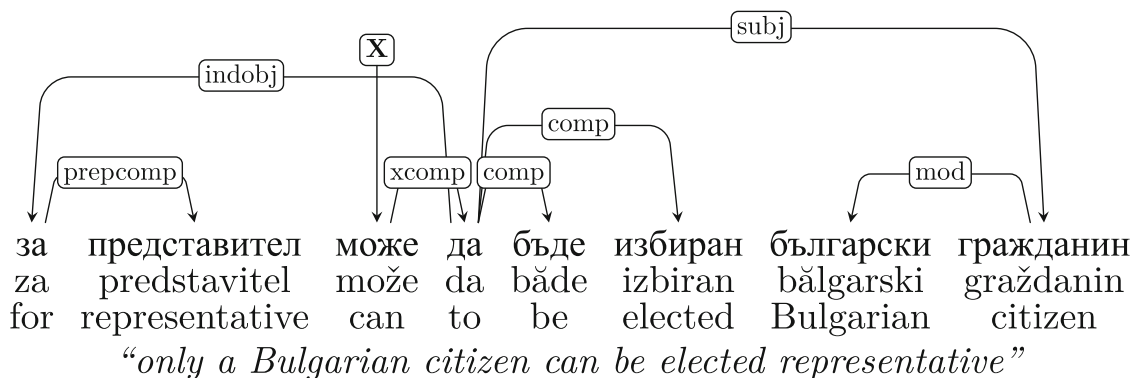


Fig. 18 Modal passive construction in [bg]: The finite modal verb (*може*) is the head, the infinitive particle (*да*) is the second-level head. The infinitive auxiliary (*бъде*) is attached to *да*, as is the passive participle of the content verb (*избран*) and the two arguments of the content verb, one of them (*за представител*) non-projectively

The head typically holds the label describing the relation of the whole verbal group to its parent. As for child nodes, subjects and negative particles are often attached to the head, especially if it is the finite element [de, en], while the arguments (objects) are attached to the content element whose valency slot they fill (often participle or infinitive). Sometimes even the subject (in [nl]) or the negative particle (in [pt]) can be attached to the non-head content element (Fig. 22). Various infinitive-marking particles (English “to”, Swedish “att”, Bulgarian “да”) are usually treated similarly to subordinating conjunctions, i.e., they either govern the infinitive [en, da, bg] or are attached to it [de, sv]. In [pt], prepositions used between the main verb and the infinitive (“*estão a usufruir*” = “*are enjoying*”) are attached to the finite verb (Fig. 24). In [bg], all modifiers of the verb including the subject are attached to the infinitive particle *да* instead of the verb below it (Fig. 18).

We intend to unify verbal groups under a common approach, but the current version 1.5 of HamleDT does not do so yet. This part is more language-dependent than the others and a further analysis is needed.

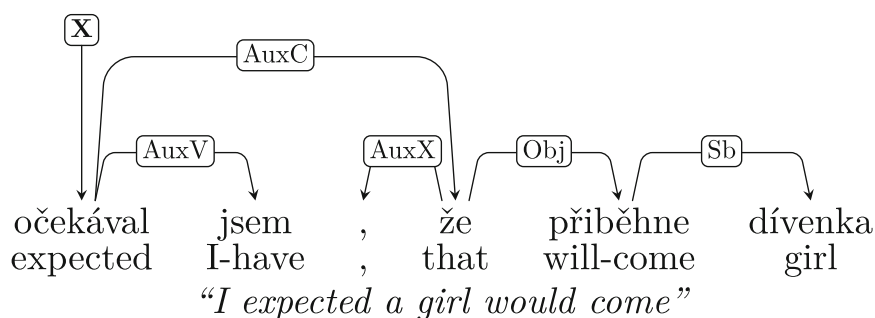


Fig. 19 Past tense in [cs]: The participle of the content verb (*očekával*) governs the finite form of the auxiliary (*jsem*). Making the auxiliary the head would cause problems because it is not always present, e. g., omitting it in this sentence would just shift the sentence to the 3rd person meaning (*He expected a girl would come*)

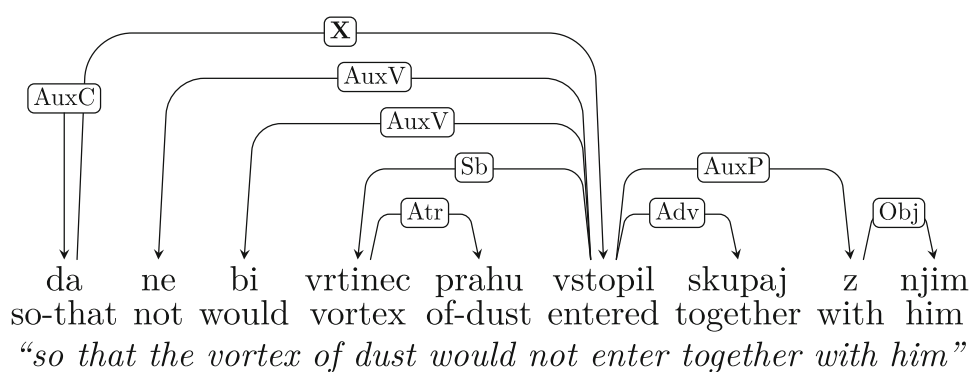


Fig. 20 Negated conditional construction in [sl]. The past participle of the content verb (*vstopil*) is the head, the negative particle (*ne*) and the auxiliary (*bi*) depend on it

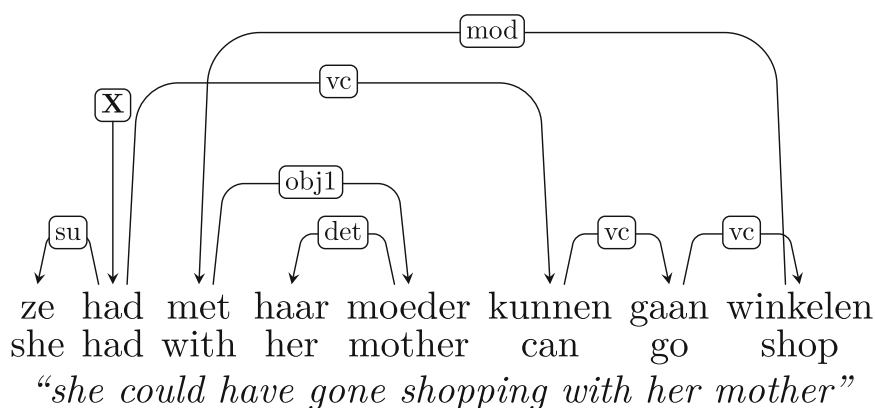


Fig. 21 Past modal construction in [nl]. The finite auxiliary verb (*had*) is the head. The subject (*ze*) is attached to the finite verb (*had*) while the modifier (*met haar moeder*) is attached non-projectively to the content verb (*winkelen*)

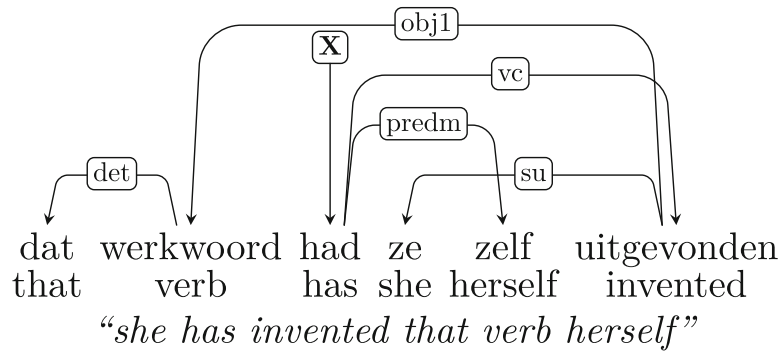


Fig. 22 Another example from [nl]. Unlike in other treebanks, even the subject (*ze*) is attached to the non-head participle (*uitgevonden*)

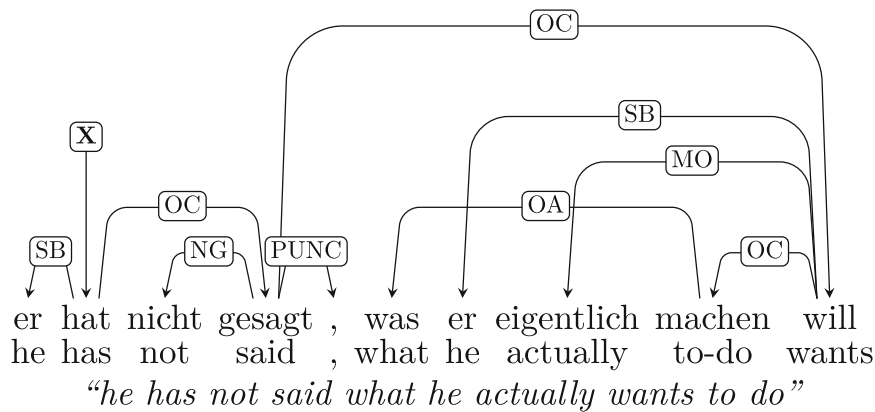


Fig. 23 A combination of perfect tense, modal verb, and infinitive in [de]. Infinitives are attached to modals as their objects in many treebanks, including [de]. The finite auxiliary verb (*hat*) is the head of the perfect tense, the participle (*gesagt*) depends on it. The subject (*er*) is attached to the finite verb (*hat*) while the object clause (*was er eigentlich machen will*) is attached to the content verb (*gesagt*)

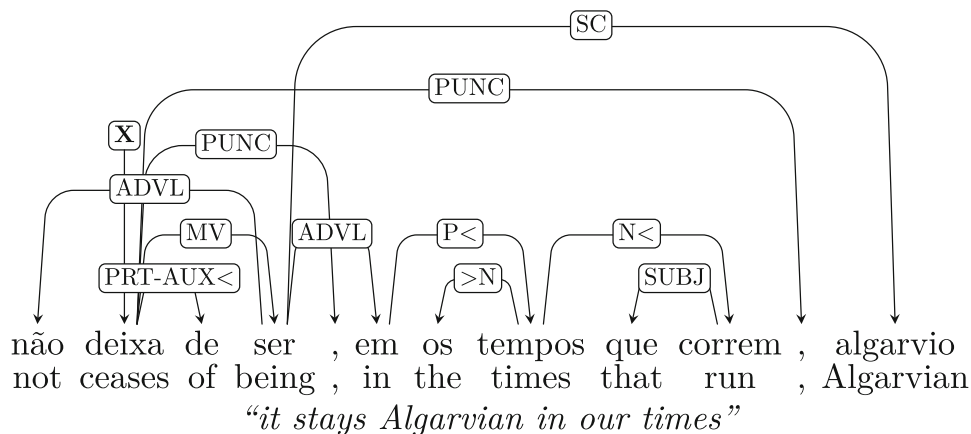


Fig. 24 Infinitive with preposition in [pt]: Preposition (*de*) is not attached between the phase verb (*deixa*) and the infinitive (*ser*). The negative particle (*não*) is attached non-projectively to the non-head verb (*ser*). Moreover, the commas around the parenthetical (*em os tempos que correm*) are also non-projective

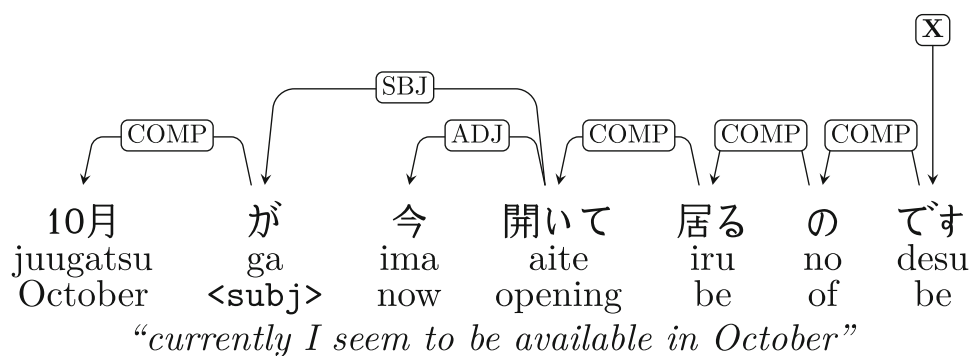


Fig. 25 [ja] *Desu* is the polite copula. *Aite* is the conjunctive form of *aku* = “to open”. The auxiliary *iru* with conjunctive of content verb together form the progressive tense. Japanese is an SOV language and left-branching structures are much preferred

Fig. 26 [fa] Note that the dependency tree of the sentence (*In mehmâni tartîb šod dâde.*) is ordered right-to-left, the way Persian is written. The analytical passive *šod dâde* is represented by a single node (token)

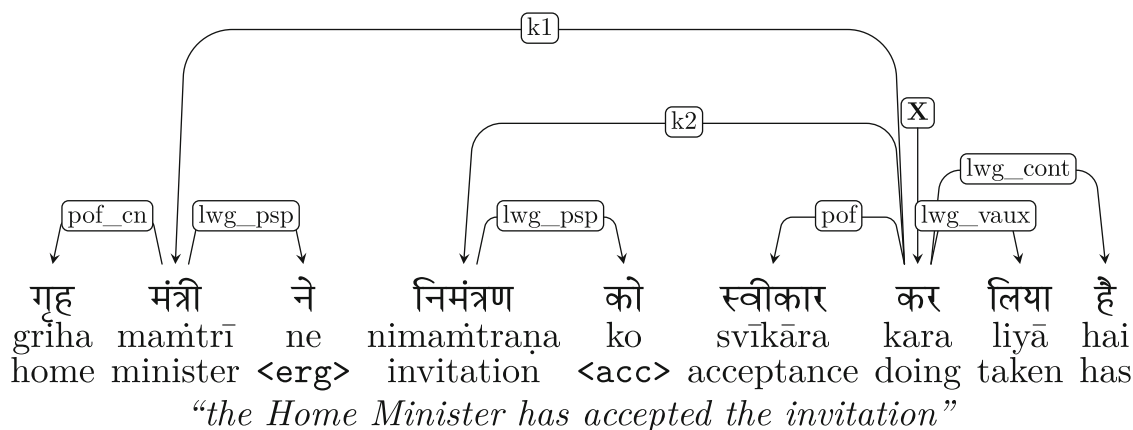
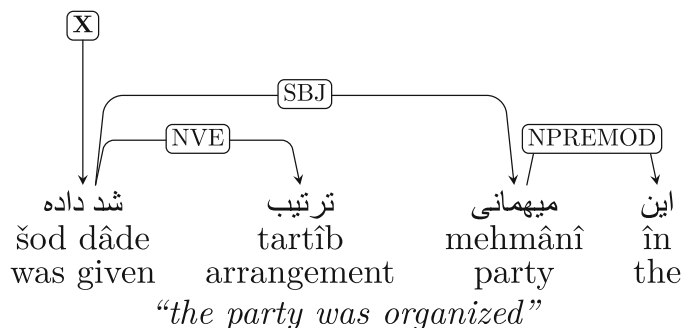


Fig. 27 [hi] *Kara* is a light verb stem, *svikāra karanā* means “to accept”. *Liyā*, the perfect participle of *lenā* “to take”, is another light verb, specifying the direction of the result of the action. *Hai* is the auxiliary verb “to be” in finite form. Content verbs govern verbal groups in the [hi] treebank; as the main verb in this case is a compound verb (*svikāra kara*), the head node of the two (*kara*) governs the whole group, even though the real content lies in the nominal element (*svikāra*)

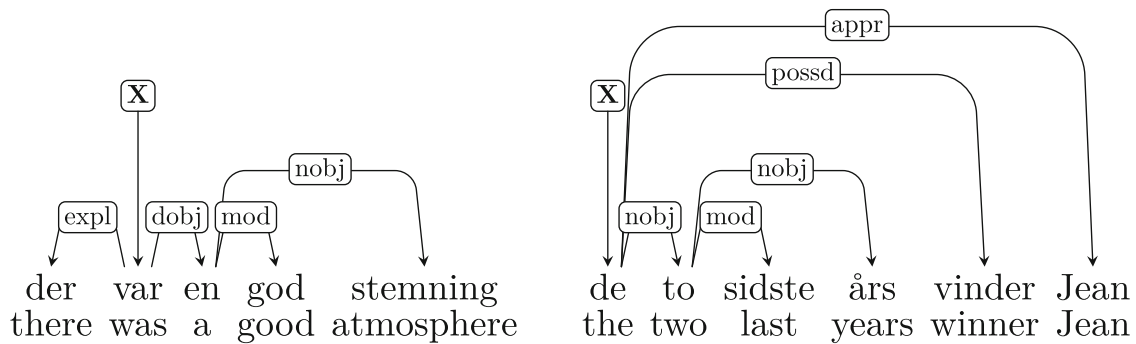


Fig. 28 Two fragments from [da] show determiners and numerals governing noun phrases

5.5 Determiner heads

The Danish treebank is probably the most extraordinary one. Nouns often depend on determiners, numerals, etc. (see Fig. 28). This approach is very rare in dependency treebanks, although it has its advocates among linguists (Hudson 2004, 2010).¹⁰

In HamleDT v1.5, we attach articles as well as other determiners to their nouns and numerals to the counted nouns.¹¹

5.6 Punctuation

Table 3 presents an overview of punctuation treatment in the treebanks. Details and exceptions are discussed below. The *type codes* at paragraph beginnings refer to the columns of the table.

Pair/Pcom: Paired punctuation marks (quotation marks, brackets, parenthesizing commas (*Pcom*) or dashes) are typically attached to the head of the segment between them. Occasionally, they are attached one level higher, to the parent of the enclosed segment, or even higher, if the parent is member of a verbal group. Attaching punctuation to higher levels may break projectivity, as in Fig. 24. The [pt] approach attaches paired punctuation to the parent of the interior segment (i.e. to the parent of the head of the segment, not to the head), unless the parent is the root or there are tokens outside the punctuation that depend on the head inside. In this latter case, the punctuation is attached to the inner head. In [tr], the *Pcom* column does not necessarily refer to *paired* punctuation; some commas are just attached to the root, which may result in non-projectivity.

Rcom: Similarly, commas before and after a relative clause are typically attached either to the root of the relative clause (be it verb or conjunction) or to its parent. In [la], the clause is sometimes headed by a subordinating conjunction, but the comma is attached to the verb below. Note, however, that a comma terminating a clause may have multiple functions: it may at the same time delimit several nested clauses, a parenthetical phrase, and/or a conjunct.

¹⁰ In Chomskian (constituency-based) approaches, it is the standard analysis that determiners function as the head of a noun phrase.

¹¹ Note however that numerals governing nouns are not restricted to [da]. Czech has a complex set of rules for numerals (motivated by the morphological agreement), which may result under some circumstances in the numeral serving as the head.

Table 3 Punctuation styles overview

Language	Fin	Pair	Pcom	Rcom	Coord	Coor1	Apos
Arabic (ar)	RN	SH	SH		HD	(PT)	
Basque (eu)	PT	PT	PT	PT		PT	PT
Bengali (bn)	(MP*)				HD		
Bulgarian (bg)	MP	SH	SH	SH	SH	SH	SH
Catalan (ca)	MP	SH	SH	SH	SH	SH	SH
Czech (cs)	RN	SH	SH	SH	HD	SH	HD
Danish (da)	MP	SH		SP/SH	PT*	SH	
Dutch (nl)	PW	PW	PW	PW		PW	PW
English (en)	MP	SH?	SP	SP	HD	SH	SP
Estonian (et)	MP	SHISP	SP	SHISP	HD	SH	SP
Finnish (fi)	MP	SH	SH	SH	SH	SH	
German (de)	MP	SH?	SP	SP	SH	PC	
Greek (el)	RN	SH			HD	SH	HD
Greek (grc)	RN	(SH)	SH	SH	HD	SH	HD
Hindi (hi)	MP	SH		(SP)		PC PT	
Hungarian (hu)	MP	SHISP	SHISP	SP	HD SP*	SP	
Italian (it)	PT	NT/PT	PT	PT		PT	PT
Japanese (ja)	MP*						
Latin (la)			SH	SH*	HD	SH	HD
Persian (fa)	MP	SH	PT	PT		PT	PT
Portuguese (pt)	MP	SP*	SP	SP	SH	SH	SP
Romanian (ro)	No punctuation						
Russian (ru)	No punctuation						
Slovene (sl)	RN	SH	SH	SH	HD	SH	HD
Spanish (es)	MP	SH	SH	SH	SH	SH	SH
Swedish (sv)	MP	NT/PT	SP	SP	PC	PC	SP
Tamil (ta)	RN	SH	SP	SP	HD	SH	
Telugu (te)	(MP)				HD		
Turkish (tr)	RR		RN*	CH	CH	CH	
HamleDT v1.5	RN	SH	SH	SH	HD	SH	SH

RN = attached to the artificial root node; RR = attached to the artificial root and serving as root for the rest of the sentence, i.e., heading the main predicate; MP = attached to the main predicate; NT = attached to the next token; PT = attached to the previous token; PW = attached to the previous word (i.e., non-punctuation token); PC = attached to the previous conjunct; SH = attached to the head of the rel. clause/subtree inside paired punc./coordination/second appos. member; SP = attached to the (grand)parent node of the rel. clause/subtree inside paired punc.; or to the first appos. member; CH = chain: attached to parent, and the head of the clause attached to the comma; for *Coord*, previous conjunct attached to comma, comma attached to next conjunct; HD = serving as head of coordination; (X) = rare in this treebank, based on very few observations; X/Y = initial X, final Y; X|Y = both observed; X? = unexplained exceptions observed; X* = see text for more details; *empty cell* = not observed

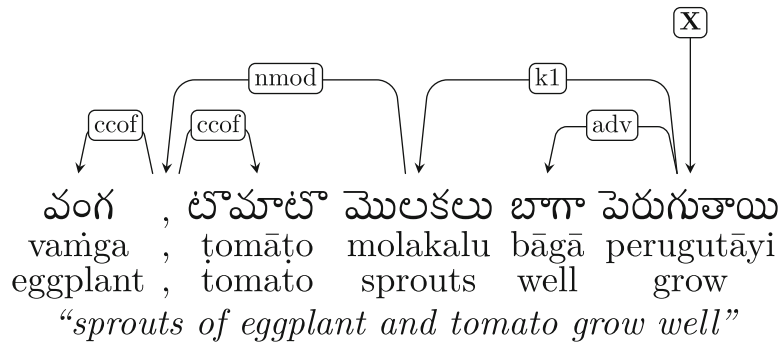
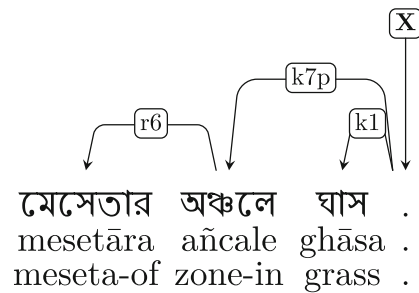


Fig. 29 Coordinating comma in [te]

Fig. 30 NULL-like usage of period in [bn]. The node with the period represents a dropped copula. Elsewhere in the treebank, such nodes are labeled by the pseudo-word-form “NULL”



“there is grass in the zone of meseta”

In several languages, commas (in [fa]) or all punctuation symbols (in [eu, it, nl]) are systematically attached to neighboring tokens.

Coord: Commas, semicolons, or dashes can also substitute coordinating conjunctions, which is important especially if the Prague style of coordination is used (see Sect. 5.1). In [te], this is the sole function of commas (see Fig. 29). In [da], which does not follow the Prague approach to coordination, we observed two adjectives modifying the same noun, separated by a comma; the comma was attached to the first “conjunct”. We list the case in the *Coord* column although the structure was not formally tagged as coordination. In [hu], coordinating commas are normally attached to the parent of the coordination. Parents that are roots of the tree are an exception: in such cases, the comma is used as the head of the coordination.

Coor1: Multi-conjunct coordination often involves one conjunction and one or more commas. Even within the same coordination family, multiple attachment schemes are possible for the commas (the previous conjunct, the head of the coordination, etc.) Additional commas are rare in [ar], where repeated conjunctions are more common.

Apos: Constructions in which two phrases describe the same object are called *appositions*. These are mostly but not solely noun phrases separated by a comma, dash, bracket, etc. as in “*Nicoletta Calzolari, the chief editor*”. Appositions are treated in the same way as parenthesis in most treebanks—the second phrase is attached to the first. Other treebanks regard appositions as coordinations—the punctuation serves as the head, with both phrases attached symmetrically.

Fin: Sentence-final punctuation (period, question mark, exclamation mark, three dots, semicolon, or colon) is attached to the artificial root node [cs, ar, sl, grc, ta], to the main predicate [bg, ca, da, de, en, es, et, fi, hu, pt, sv], or to the previous token

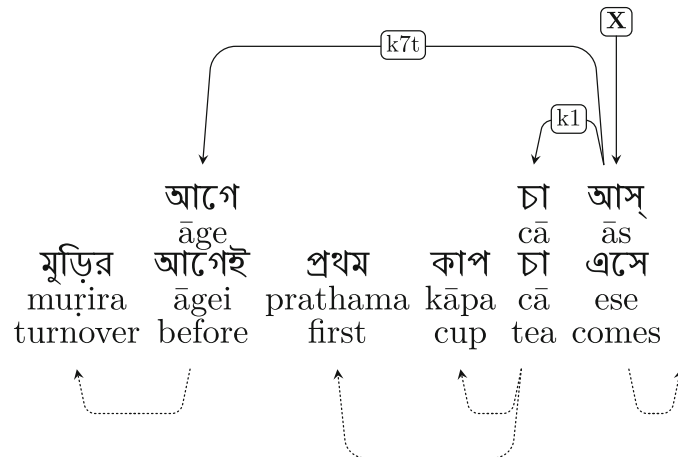


Fig. 31 *The first cup of tea comes before the turnover.* [bn] captures dependencies between chunks, not between tokens. Every sentence has been chunked and chunk headwords serve as nodes of the tree (their word forms are replaced by lemmas). The dotted dependencies below the sentence indicate which tokens belong to which chunk. Neither these dependencies nor the chunk-dependent words are visible in the treebank. The original sentences cannot be reconstructed from the trees

[eu, it, ja, nl].¹² In [la, ro, ru], there is no final punctuation. It is also extremely rare in [bn, te]; however, there are a few punctuation nodes in [bn] that govern other nodes in the sentence. In fact, these nodes actually should have been labeled NULL to represent a copula or other constituents missing from the surface (Fig. 30). Such NULL nodes appear elsewhere in [bn]. Punctuation is attached to the artificial root node in [tr] but instead of being a sibling of the main predicate, it governs the predicate. Note that some languages (e.g. Czech) may require final quotation marks (if present) to appear after the final period, but in [cs], it is not treated as final punctuation (unlike the period). Such quotation marks may end up attached non-projectively to the main verb.

A few treebanks [bg, cs, la, sl] use separate nodes for periods that mark abbreviations and ordinal numbers. These nodes are attached to the previous node (i. e., the abbreviation). In [cs], this rule has a higher priority even in cases where a period serves as an abbreviation marker and a sentence terminator at the same time. Most other treebanks are tokenized so that the period shares a node with the abbreviation (see also Sect. 5.7).

In HamleDT v1.5, we treat apposition as parenthesis, we attach paired punctuation to the root of the subtree inside and sentence-final punctuation to the artificial root node, mostly for consistency reasons. For the other punctuation types, a further analysis is needed.

5.7 Tokenization and sentence segmentation

The only aspect that remains unchanged in HamleDT is tokenization and segmentation. Our harmonized trees always have the same number of nodes and sentences as the original annotation, despite some variability in the approaches we observe in the original treebanks.

¹² In [ja], the previous token essentially means the main predicate, but if it is followed by a question particle then the punctuation node is attached to the particle.

5.7.1 Multi-word expressions and missing tokens

Some treebanks collapse multi-word expressions into single nodes [ca, da, es, eu, fa, hu, it, nl, pt, ro, ru]. Collapsing is restricted to personal names in [hu] and to named entities in [ro]. In [fa], it is used for analytical verb forms. The word form of the node is composed of all the participating words, joined by underscore characters or even by spaces [fa].

In [bn, te], dependencies are annotated between chunks instead of words (Fig. 31). Therefore, one node may represent a whole noun phrase with modifiers and postpositions. The treebank only shows chunk headwords, which means we cannot reconstruct the original sentence. On a similar note, punctuation tokens have been deleted from two treebanks ([ro, ru]; see also Sect. 5.6).

5.7.2 Split tokens

On the other hand, orthographic words may be split into syntactically autonomous parts in some treebanks [ar, fa]. For instance, the Arabic word *وبالفالوجة* (*wabiālfālūjah* = “and in al-Falujah”) is separated into *wa/CONJ* + *bi/PREP* + *AlfAlwjp/NOUN_PROP*. In [ta], the suffix *-um* indicating a coordination is treated as a separate token (see Sect. 5.1; Fig. 8).

5.7.3 Artificial nodes

Occasionally [bn, hi, te, ru], we see an inserted NULL node, which mostly stands for participants deleted on the surface, e.g., copulas [bn, ru] or conjuncts as in the Hindi example in Fig. 32.

Along the same lines, some treebanks of pro-drop languages [ca, es] use empty nodes (with artificial word “_”) representing missing subjects, as in the following Spanish sentence: “_ *Afirmó que _ sigue el criterio europeo y que _ trata de incentivar el mercado donde no lo hay.*” = “He said he follows the European standard and encourages the market where there is none.” All the underscores mark subjects of the following verbs and could be translated as “he”.

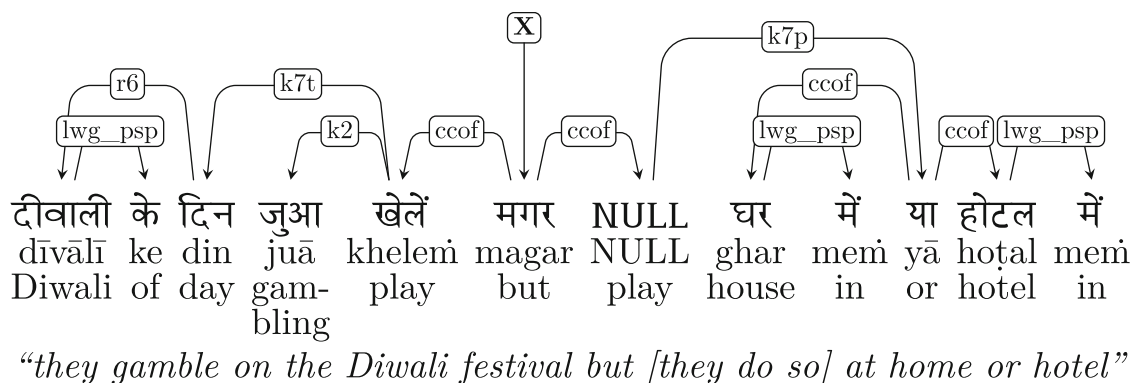


Fig. 32 A NULL node for a deleted verb (serving as head of conjunct) in [hi]

Underscore/NULL nodes also appear in [tr], where they encode additional information related to morphological derivation.

5.7.4 Sentence segmentation

Similarly to tokenization, we also treat sentence segmentation as fixed, despite some less usual solutions: in [ar], sentence-level units are paragraphs rather than sentences, which explains the high average sentence length in Table 1. In contrast, [ro] annotates every clause as a separate tree.

6 Obtaining HamleDT

Twelve harmonized treebanks from HamleDT v1.5 [ar, cs, da, fa, fi, grc, la, nl, pt, ro, sv, ta] are directly available for download from our web site:

<http://ufal.mff.cuni.cz/hamledt>.

The license terms of the rest of the treebanks prevent us from redistributing them directly (in their original or normalized form), but most of them are easily acquirable for research purposes, under the links given in Appendix 1). We provide the software that can be used to normalize and display the data after obtaining them from the original provider.

All the normalizations are implemented in Treex (formerly TectoMT) (Popel and Žabokrtský 2010), a modular open-source framework for structured language processing, written in Perl.¹³ In addition to normalization scripts for each treebank, Treex contains also other transformations, so for example, coordinations in any treebank can be converted from Prague to Stanford style.

The tree editor TrEd¹⁴ can open Treex files and display original and normalized trees side-by-side on multiple platforms.

7 Conclusion

We provide a thorough analysis and discussion of varying annotation approaches to a number of syntactic phenomena, as they appear in publicly available treebanks, for many languages.

We propose a method for automatic normalization of the discussed annotation styles. The method applies transformation rules conditioned on the original structural annotation, dependency labels and morphosyntactic tags. We also propose unification of the tag sets for parts of speech, morphosyntactic features, and dependency relation labels. We take care to make the structural transformations and the morphosyntactic tagset unification as reversible as possible.¹⁵

¹³ <http://ufal.mff.cuni.cz/treex/>.

¹⁴ <http://ufal.mff.cuni.cz/tred/> with EasyTreex extension.

¹⁵ We do not attempt at reversibility when unifying dependency relations.

We provide an implementation of the transformations in the Treex NLP framework. Treex can also be used for transforming the data to other annotation styles besides the one we propose (cf. Popel et al. 2013). The resulting collection of harmonized treebanks, called HamleDT v1.5, is available to the research community according to the original licenses. A subset of the treebanks whose license terms permit redistribution is available directly from us. For the rest, users need to acquire the original data and apply our transformation tool.

Several future directions of our work are possible. Besides deepening the current level of harmonization (especially for verbal groups), we plan on adding new treebanks and languages, for which resources exist (e.g., French, Hebrew, Chinese, Icelandic, Ukrainian or Georgian). We also want to run parsing experiments and evaluate the various annotation styles from the point of view of learnability by parsers.

Acknowledgments The authors wish to express their gratitude to all the creators and providers of the respective corpora. The work on this project was supported by the Czech Science Foundation Grant Nos. P406/11/1499 and P406/14/06548P, by the European Union Seventh Framework Programme under Grant Agreement FP7-ICT-2013-10-610516 (QTLep), and by research resources of the Charles University in Prague (PRVOUK). This work has been using language resources developed and/or stored and/or distributed by the LINDAT/CLARIN project of the Ministry of Education of the Czech Republic (Project LM2010013). Finally, we are very grateful for the numerous valuable comments provided by the anonymous reviewers.

Appendix 1: List of included languages and treebanks

- Arabic [ar]: Prague Arabic Dependency Treebank 1.0/CoNLL 2007 (Smrž et al. 2008)
<http://padt-online.blogspot.com/2007/01/conll-shared-task-2007.html>
- Basque [eu]: Basque Dependency Treebank, a larger version than the one included in CoNLL 2007, generously provided by IXA Group (Aduriz et al. 2003)
<http://hdl.handle.net/10230/17098>
- Bengali [bn], Hindi [hi] and Telugu [te]: Hyderabad Dependency Treebank/ICON 2010 (Husain et al. 2010)
<http://ltrc.iiit.ac.in/icon/2010/nlptools/>
- Bulgarian [bg]: BulTreeBank (Simov and Osenova 2005)
<http://www.bultreebank.org/indexBTB.html>
- Catalan [ca] and Spanish [es]: AnCora (Taulé et al. 2008)
<http://clic.ub.edu/corpus/en/ancora-descarregues>
- Czech [cs]: Prague Dependency Treebank 2.0/CoNLL 2009 (Hajič et al. 2006)
<http://ufal.mff.cuni.cz/pdt2.0/>
- Danish [da]: Danish Dependency Treebank/CoNLL 2006 (Kromann et al. 2004), now part of the Copenhagen Dependency Treebank
<http://code.google.com/p/copenhagen-dependency-treebank/>
- Dutch [nl]: Alpino Treebank/CoNLL 2006 (van der Beek et al. 2002)
<http://odur.let.rug.nl/~vannoord/trees/>
- English [en]: Penn TreeBank 3/CoNLL 2007 (Marcus et al. 1993)
<http://www.cis.upenn.edu/~treebank/>

- Estonian [et]: Eesti keele puudepank/Arborest (Bick et al. 2004)
<http://www.cs.ut.ee/~kaili/Korpus/puud/>
- Finnish [fi]: Turku Dependency Treebank (Haverinen et al. 2010)
<http://bionlp.utu.fi/fintreebank.html>
- German [de]: Tiger Treebank/CoNLL 2009 (Brants et al. 2004)
<http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html>
- Greek (modern) [el]: Greek Dependency Treebank (Prokopidis et al. 2005)
<http://gdt.ilsp.gr/>
- Greek (ancient) [grc] and Latin [la]: Ancient Greek and Latin Dependency Treebanks (Bamman and Crane 2011)
<http://nlp.perseus.tufts.edu/syntax/treebank/greek.html>,
<http://nlp.perseus.tufts.edu/syntax/treebank/latin.html>
- Hindi [hi]: *see Bengali*
- Hungarian [hu]: Szeged Treebank (Csendes et al. 2005)
http://www.inf.u-szeged.hu/projectdirs/hlt/index_en.html
- Italian [it]: Italian Syntactic-Semantic Treebank/CoNLL 2007 (Montemagni et al. 2003)
<http://medialab.di.unipi.it/isst/>
- Japanese [ja]: Verbmobil (Kawata and Bartels 2000)
<http://www.sfs.uni-tuebingen.de/en/tuebajs.shtml>
- Latin [la]: *see Greek (ancient)*
- Persian [fa]: Persian Dependency Treebank (Rasooli et al. 2011)
<http://dadegan.ir/en/persiandependencytreebank>
- Portuguese [pt]: Floresta sintá(c)tica (Afonso et al. 2002)
http://www.linguateca.pt/floresta/info_floresta_English.html
- Romanian [ro]: Romanian Dependency Treebank (Călăcean 2008)
<http://www.phobos.ro/roric/texts/xml/>
- Russian [ru]: Syntagrus (Boguslavsky et al. 2000)
<http://ruscorpora.ru/en/>
- Slovene [sl]: Slovene Dependency Treebank/CoNLL 2006 (Džeroski et al. 2006)
<http://nl.ijs.si/sdt/>
- Spanish [es]: *see Catalan*
- Swedish [sv]: Talbanken05 (Nilsson et al. 2005)
<http://www.msi.vxu.se/users/nivre/research/Talbanken05.html>
- Tamil [ta]: TamilTB (Ramasamy and Žabokrtský 2012)
<http://ufal.mff.cuni.cz/~ramasamy/tamiltb/0.1/>
- Telugu [te]: *see Bengali*
- Turkish [tr]: METU-Sabancı Turkish Treebank (Atalay et al. 2003)
<http://ii.metu.edu.tr/corpus/>

Appendix 2: Examples of harmonization of dependency relations

See Tables 4 and 5.

Table 4 The Bengali treebank [bn] uses 42 dependency labels, but we show only 12 most frequent ones

Orig. label	Tokens	Distribution of HamleDT v1.5 labels
k1	1,168	Sb = 98 % Coord = 2 %
main	1,130	Pred = 85 % Coord = 14 %
r6	790	Atr = 98 % Coord = 2 %
k2	788	Obj = 95 % Coord = 5 %
ccof	602	Pred = 40 % Atr = 23 % Obj = 12 % Coord = 8 % Adv = 8 % Sb = 7 % Pnom = 2 %
vmod	583	Adv = 98 % Coord = 2 %
pof	421	Obj = 100 %
k7p	325	Adv = 98 % Coord = 2 %
k7t	303	Adv = 100 %
nmod	233	Atr = 91 % Coord = 9 %
k1s	202	Pnom = 98 % Coord = 2 %
k7	152	Adv = 97 % Coord = 3 %
other	470	Atr = 41 % Adv = 37 % Obj = 11 % Coord = 4 % Sb = 4 % Atv = 2 % rest < 0.5 %

The remaining 30 labels are summarized in the last line. Bengali dependency labels are explained in <http://trc.iit.ac.in/nlptools2010/files/documents/dep-tagset.pdf> and their mapping to HamleDT v1.5 dependency labels is relatively straightforward, except for coordinations, where the Bengali treebank marks the conjunction with the dependency relation, while in HamleDT v1.5, the conjuncts are marked with the dependency relation and the conjunction is marked with *Coord*. For example, k7p is *location in space*, k7t *location in time* and k7 *location elsewhere*; all three labels are basically mapped to *Adv* (adverbial)

Table 5 The English treebank [en] (from CoNLL 2007) uses 20 dependency labels, but their mapping to HamleDT v1.5 labels is not straightforward

Orig. label	Tokens	Distribution of HamleDT v1.5 labels
NMOD	155,951	Atr = 62 % AuxA = 22 % AuxP = 13 % Coord = 1 % AuxV = 1 % rest = 1 %
P	52,051	AuxX = 44 % AuxK = 36 % AuxG = 20 % rest < 0.5 %
PMOD	45,207	Adv = 50 % Atr = 40 % Coord = 6 % AuxP = 2 % NR = 1 % Obj = 1 % rest < 0.5 %
SBJ	35,446	Sb = 94 % Coord = 2 % NR = 1 % Atr = 1 % Obj = 1 % Adv = 1 % rest < 0.5 %
ADV	32,202	AuxP = 56 % Adv = 30 % AuxC = 5 % NR = 3 % Atr = 2 % AuxV = 2 % Obj = 1 % Coord = 1 % rest < 0.5 %
OBJ	30,507	Obj = 55 % Adv = 29 % AuxV = 8 % Coord = 5 % Atr = 2 % rest = 1 %
COORD	22,865	Atr = 34 % Adv = 21 % Obj = 12 % Pred = 11 % Sb = 8 % AuxV = 5 % Phom = 3 % AuxP = 2 % NR = 2 % Coord = 1 % AuxA = 1 % rest = 1 %
VMOD	21,053	AuxV = 30 % AuxC = 24 % Phom = 20 % Neg = 10 % Adv = 6 % Atr = 3 % Coord = 2 % Obj = 2 % NR = 2 % AuxP = 1 % Sb = 1 % rest < 0.5 %
ROOT	18,791	Pred = 69 % AuxV = 18 % Coord = 8 % ExD = 4 % rest < 0.5 %
AMOD	15,269	Atr = 52 % Adv = 19 % AuxP = 14 % NR = 9 % AuxC = 3 % AuxV = 1 % rest = 1 %
VC	13,745	Pred = 29 % Adv = 25 % Obj = 23 % AuxV = 10 % Atr = 9 % Coord = 2 % NR = 1 % rest < 0.5 %
IOBJ	1,883	Obj = 92 % Adv = 3 % Coord = 2 % Atr = 1 % Sb = 1 % rest < 0.5 %
CC	1,336	NR = 98 % Neg = 2 %
PRT	1,268	AuxV = 95 % AuxC = 2 % Adv = 2 % rest < 0.5 %
PRN	1,259	Atr = 43 % Adv = 27 % AuxP = 8 % NR = 7 % Obj = 6 % Coord = 5 % AuxV = 1 % AuxC = 1 % rest < 0.5 %
LGS	1,211	AuxP = 99 % AuxC = 1 % rest < 0.5 %
DEP	892	AuxP = 46 % Atr = 23 % Adv = 10 % NR = 9 % Neg = 4 % AuxC = 3 % AuxA = 2 % Coord = 1 % rest = 1 %
GAP	272	Atr = 47 % AuxP = 38 % Adv = 10 % NR = 2 % Coord = 1 % AuxC = 1 % Neg = 1 %
EXP	219	Adv = 84 % AuxV = 11 % Coord = 5 %
TMP	149	Atr = 97 % NR = 3 %

In practice, we found the English CoNLL 2007 labels not helpful, and we based the conversion only on dependency structure and morphological tags

Appendix 3: List of dependency relation labels in figures

Language	Label	Description	Example
	X	Our meta-label that represents the unknown relation of the depicted subtree to its unshown parent	
bg	comp	Complement, i.e. argument of non-verbal head, non-finite verbal head, copula	Figure 18
bg	indobj	Child is indirect object of parent	Figure 18
bg	mod	Child is modifier, e.g. of a noun phrase, or a negative particle modifying a verb etc.	Figure 18
bg	prepcomp	Child is noun phrase, parent is preposition	Figure 18
bg	subj	Child is subject of parent	Figure 18
bg	xcomp	Child is clausal complement; this includes complements of modal verbs	Figure 18
ca	CO	Child is coordinating conjunction, parent is the first conjunct	Figure 4
ca	CONJUNCT	Parent is the first conjunct, child is one of the other conjuncts	Figure 4
ca	PUNC	Child is punctuation symbol	Figure 4
cs, sl, la, ta	Adv	Child is adverbial modifier of parent	Figure 2
cs, sl, la, ta	Atr	Parent is noun, child is its attribute	Figure 9
cs, sl, la, ta	AuxC	Child is subordinating conjunction, parent is governing predicate. The relation of the subordinate clause to the parent is labeled at the grandchild	Figure 19
cs, sl, la, ta	AuxP	Child is preposition. The relation of the prepositional phrase to the parent is labeled at the grandchild	Figure 2
cs, sl, la, ta	AuxV	Child is auxiliary verb or negative particle, parent is content verb	Figure 19
cs, sl, la, ta	AuxX	Child is comma and does not serve as coordination root	Figure 2
cs, sl, la, ta	AuxZ	Emphasizing word	Figure 8
cs, sl, la, ta	Coord	Child serves as root of a coordinate structure	Figure 1
cs, sl, la, ta	Obj	Child is object of parent	Figure 2
cs, sl, la, ta	Pred	Child is predicate of a main clause	Figure 2
cs, sl, la, ta	Sb	Child is subject of parent	Figure 19

Language	Label	Description	Example
cs, ta	_M	Suffix to a label, saying that the child is a conjunct. The main label tags its relation to the parent of the coordinate structure	Figure 1
da	appr	Restrictive apposition (no comma)	Figure 28
da	conj	Child is conjunct, parent is first conjunct or coordinating conjunction	Figure 6
da	coord	Parent is conjunct, child is coordinating conjunction	Figure 6
da	dobj	Child is direct object of parent	Figure 28
da	expl	Child is expletive subject of parent	Figure 28
da	mod	Modifier, e.g. attribute of noun, adverbial modifier of verb, adjective attached to determiner etc.	Figure 28
da	nobj	Child is noun phrase or infinitive, parent is e.g. determiner, numeral, preposition etc.	Figure 28
da	punct	Child is punctuation symbol	Figure 6
da	possd	Child is argument of possessive parent, i.e. child is the thing possessed	Figure 28
de	CD	Child is coordinating conjunction, parent is one conjunct and right sibling is the other conjunct	Figure 3
de	CJ	Parent and child are conjuncts	Figure 3
de	MO	Modifier. In NPs only focus particles are annotated as modifiers	Figure 23
de	NG	Child is negative particle, parent is negated verb	Figure 23
de	NK	Noun Kernel. Child attached within a noun phrase or a prepositional phrase	Figure 10
de	OA	Child is accusative object of parent	Figure 23
de	OC	Clausal object. Also verb tokens building a complex verbal form and modal constructions	Figure 23
de	PUNC	Child is punctuation symbol	Figure 3
de	SB	Child is subject of parent	Figure 23
es	atr	Attribute. E.g. child is adverbial/prepositional phrase, parent is verb	Figure 12
es	cd	Child is direct object of parent	Figure 12
es	conj	Child is subordinating conjunction	Figure 12
es	s.a	Child is adjectival phrase, parent is not verb	Figure 12
es	sn	Child is noun phrase. Parent may be e.g. preposition	Figure 12
es	spec	Specifier. E.g. child is determiner and parent is noun	Figure 12

Language	Label	Description	Example
es	suj	Child is subject of parent	Figure 12
fa	NPREMOD	Child is premodifier of parent noun	Figure 26
fa	NVE	Child is non-verbal element of compound verb. Parent is verbal element	Figure 26
fa	SBJ	Child is subject of parent	Figure 26
hi	lwg_cont	Child is additional node of a complex expression; child and parent together perform certain function	Figure 27
hi	lwg_psp	Child is postposition and modifies a noun	Figure 11
hi	lwg_vaux	Child is auxiliary verb, parent is content verb	Figure 27
hi	pof	Part of relation, e.g. part of conjunct verb	Figure 27
hi	pof_cn	Part of relation	Figure 27
hi, bn, te	adv	Child is adverbial modifier (only adverbs of manner) of parent	Figure 29
hi, bn, te	ccof	Child is conjunct, parent is coordinating conjunction or comma	Figure 29
hi, bn, te	k1	Child is karta (doer/agent/subject) of parent predicate	Figure 27
hi, bn, te	k2	Child is karma (patient/object) of parent predicate	Figure 27
hi, bn, te	k7p	Child is deshadhikarana (location in space) of the parent predicate	Figure 30
hi, bn, te	k7t	Child is kaalaadhikarana (location in time) of the parent predicate	Figure 31
hi, bn, te	nmod	Parent is noun, child is its attribute	Figure 29
hi, bn, te	nmod_adj	Child is adjective and modifies a noun	Figure 11
hi, bn, te	r6	Shashthi (possessive). Child is possessor in genitive, parent is the possessed noun	Figure 30
hu	ATT	Attribute	Figure 15
hu	CONJ	Child is conjunction (coordinating or subordinating)	Figure 5
hu	DET	Child is determiner, parent is noun	Figure 15
hu	ILL	Child is verbal argument in illative case	Figure 15
hu	OBJ	Child is object of parent	Figure 15
hu	PUNCT	Child is punctuation symbol	Figure 5
hu	SUBJ	Child is subject of parent	Figure 15

Language	Label	Description	Example
it	cong_sub	Parent is subordinating conjunction	Figure 13
it	det	Child is determiner, parent is noun	Figure 13
it	modal	Child is modal (dovere, volere, potere) or aspectual (andare, venire, stare) verb, parent is content verb	Figure 13
it	pred	Parent is verb (often it is copula), child is predicative complement (nominal predicate)	Figure 13
it	sogg	Child is subject of parent	Figure 13
ja	ADJ	Child is adjunct of parent	Figure 25
ja	COMP	Complement, e.g. verb attached to another verb form, noun attached to postposition etc.	Figure 25
ja	SBJ	Child is subject of parent	Figure 25
nl	det	Child is determiner, parent is noun	Figure 21
nl	mod	Child is adverbial modifier (bijwoordelijke bepaling) of parent	Figure 21
nl	obj1	Child is direct object; this includes nouns attached to prepositions!	Figure 21
nl	predm	Child determines state (adverbial modifier), parent is predicate	Figure 22
nl	su	Child is subject of parent	Figure 21
nl	vc	Verbal complement. Example: parent is modal, child is infinitive	Figure 21
pt	>N	Child is left dependent of nominal core	Figure 24
pt	ADVL	Child is adverbial adjunct (adjunto adverbial) of parent	Figure 24
pt	MV	Child is main verb, parent may be e.g. modal verb	Figure 24
pt	N <	Child is right dependent of nominal core	Figure 24
pt	P <	Child is right dependent of preposition	Figure 24
pt	PRT-AUX <	Child is verbal particle (partícula de ligação verbal), e.g. between modal and content verb, parent would be modal	Figure 24
pt	PUNC	Child is punctuation symbol	Figure 24
pt	SC	Child is nominal predicate (predicativo do sujeito), parent is copula	Figure 24
pt	SUBJ	Child is subject of parent	Figure 24
ro	rel.conj.	Parent is coordinating conjunction, child is conjunct	Figure 7
ru	1-KOMIII	Child is argument other than subject. Also: genitive noun modifier of another noun	Figure 17

Language	Label	Description	Example
ru	агент	Child is agent-object of passive parent	Figure 17
ru	опред	Parent is noun, child is its attribute	Figure 17
ru	пасс-анал	Child is passive participle, parent is finite auxiliary verb	Figure 17
ru	предик	Parent is predicate, child is subject	Figure 17
ta	AComp	Child is (obligatory) adverbial complement of parent	Figure 8
tr	ОБЪЕКТ	Child is object of parent	Figure 16
tr	QUESTION PARTICLE	Child is question particle, parent is verb	Figure 16
tr	SUBJECT	Child is subject of parent	Figure 16
tr	VOCATIVE	Child is vocative noun phrase serving as doer (actor) of parent verb	Figure 16

References

- Aduriz, I., Aranzabe, M. J., Arriola, J. M., Atutxa, A., Díaz de Ilarraza, A., Garmendia, A., & Oronoz, M. (2003). Construction of a Basque dependency treebank. In *Proceedings of the 2nd workshop on treebanks and linguistic theories*.
- Afonso, S., Bick, E., Haber, R., & Santos, D. (2002). “Floresta sintá(c)tica”: A treebank for Portuguese. In *Proceedings of the 3rd international conference on language resources and evaluation (LREC)* (pp. 1968–1703).
- Atalay, N. B., Oflazer, K., Say, B., & Inst, I. (2003). The annotation process in the Turkish treebank. In *Proceedings of the 4th international workshop on linguistically interpreted corpora (LINC)*.
- Bamman, D., & Crane, G. (2011). The ancient Greek and Latin dependency treebanks. In C. Sporleder, A. Bosch, & K. Zervanou (Eds.), *Language technology for cultural heritage, theory and applications of natural language processing* (pp. 79–98). Berlin, Heidelberg: Springer.
- Bengoetxea, K., & Gojenola, K. (2009). Exploring treebank transformations in dependency parsing. In *Proceedings of the international conference RANLP-2009. Borovets, Bulgaria* (pp. 33–38). Association for Computational Linguistics.
- Bharati, A., Chaitanya, V., & Sangal, R. (1994). *Natural language processing: A paninian perspective*. New Delhi: Prentice-Hall of India.
- Bick, E., Uibo, H., & Müürisep, K. (2004). Arborest—A VISL-style treebank derived from an Estonian constraint grammar corpus. In *Proceedings of treebanks and linguistic theories*.
- Boguslavsky, I., Grigorieva, S., Grigoriev, N., Kreidlin, L., & Frid, N. (2000). Dependency treebank for Russian: Concept, tools, types of information. In *Proceedings of the 18th conference on computational linguistics* (Vol. 2, pp. 987–991).
- Bosco, C., Montemagni, S., Mazzei, A., Lombardo, V., Lenci, A., Lesmo, L., Attardi, G., Simi, M., Lavelli, A., Hall, J., Nilsson, J., & Nivre, J. (2010). Comparing the influence of different treebank annotations on dependency parsing.
- Brants, S., Dipper, S., Eisenberg, P., Hansen, S., König, E., Lezius, W., et al. (2004). TIGER: Linguistic interpretation of a German corpus. *Journal of Language and Computation*, 2(4), 597–620. Special Issue.
- Buchholz, S., & Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL* (pp. 149–164).
- Călăcean, M. (2008). Data-driven dependency parsing for Romanian. Master’s thesis, Uppsala University.
- Civit, M., Martí, M. A., & Buří, N. (2006). Cat3LB and Cast3LB: From constituents to dependencies. In T. Salakoski, F. Ginter, S. Pyysalo, & T. Pahikkala (Eds.), *FinTAL*, Vol. 4139 of *Lecture notes in computer science* (pp. 141–152). Berlin: Springer.
- Csendes, D., Csirik, J., Gyimóthy, T., & Kocsor, A. (2005). The Szeged treebank. In V. Matoušek, P. Mautner, & T. Pavelka (Eds.), *TSD*, Vol. 3658 of *Lecture notes in computer science* (pp. 123–131). Berlin: Springer.
- de Marneffe, M.-C., & Manning, C. D. (2008). Stanford typed dependencies manual.
- Džeroski, S., Erjavec, T., Ledinek, N., Pajas, P., Žabokrtský, Z., & Žele, A. (2006). Towards a slovene dependency treebank. In *Proceedings of the fifth international language resources and evaluation conference, LREC 2006. Genova, Italy* (pp. 1388–1391). European Language Resources Association (ELRA).
- Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., & Zhang, Y. (2009). The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th conference on computational natural language learning (CoNLL-2009)*, June 4–5. Boulder, Colorado, USA.
- Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M., Žabokrtský, Z., & Ševčíková-Razímová, M. (2006). Prague dependency treebank 2.0. CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia.
- Haverinen, K., Viljanen, T., Laippala, V., Kohonen, S., Ginter, F., & Salakoski, T. (2010). Treebanking finnish. In M. Dickinson, K. Müürisep, & M. Passarotti (Eds.), *Proceedings of the ninth international workshop on treebanks and linguistic theories (TLT9)* (pp. 79–90).
- Hudson, R. (2004). Are determiners heads? *Functions of Language*, 11(1).
- Hudson, R. (2010). An encyclopedia of word grammar and English grammar. London, UK: University College London. <http://tinyurl.com/wg-encyc>.

- Husain, S., Mannem, P., Ambati, B., & Gadde, P. (2010). The ICON-2010 tools contest on Indian language dependency parsing. In *Proceedings of ICON-2010 tools contest on Indian language dependency parsing*. Kharagpur, India.
- Hwa, R., Resnik, P., Weinberg, A., Cabezas, C. I., & Kolak, O. (2005). Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(3), 311–325.
- Kawata, Y., & Bartels, J. (2000). Stylebook for the Japanese treebank in verbmobil. In *Report 240*. Tübingen, Germany.
- Kromann, M. T., Mikkelsen, L., & Lyng, S. K. (2004). Danish dependency treebank.
- Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2), 313–330.
- Mareček, D., & Žabokrtský, Z. (2012). Exploiting reducibility in unsupervised dependency parsing. In *Proceedings of EMNLP-CoNLL'12* (pp. 297–307).
- McDonald, R., Nivre, J., Quirnbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Castelló, N. B., & Lee, J. (2013). Universal dependency annotation for multilingual parsing. In *Proceedings of the ACL 2013*. Association for Computational Linguistics.
- McDonald, R., Petrov, S., & Hall, K. (2011a). Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 62–72). Stroudsburg, PA, USA. Association for Computational Linguistics.
- McDonald, R., Petrov, S., & Hall, K. (2011b). Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 62–72). Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Mel'čuk, I. A. (1988). *Dependency syntax: Theory and practice*. New York: State University of New York Press.
- Montemagni, S., Barsotti, F., Battista, M., Calzolari, N., Corazzari, O., Lenci, A., et al. (2003). Building the Italian syntactic-semantic treebank. In A. Abeillé (Ed.), *Building and using parsed corpora* (pp. 189–210). Dordrecht: Kluwer.
- Nilsson, J., Hall, J., & Nivre, J. (2005). MAMBA Meets TIGER: Reconstructing a Swedish treebank from antiquity. In *Proceedings of the NODALIDA special session on treebanks*.
- Nilsson, J., Nivre, J., & Hall, J. (2006). Graph transformations in data-driven dependency parsing. In *Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the association for computational linguistics* (pp. 257–264).
- Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., & Yuret, D. (2007). The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL 2007 shared task. Joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.
- Popel, M., & Žabokrtský, Z. (2010). TectoMT: Modular NLP framework. In *Advances in natural language processing* (pp. 293–304).
- Popel, M., Mareček, D., Štěpánek, J., Zeman, D., & Žabokrtský, Z. (2013). Coordination structures in dependency treebanks. In *Proceedings of the 51st annual meeting of the association for computational linguistics* (pp. 517–527). Sofia, Bulgaria. Association for Computational Linguistics.
- Prokopidis, P., Desipri, E., Koutsombogera, M., Papageorgiou, H., & Piperidis, S. (2005). Theoretical and practical issues in the construction of a Greek dependency treebank. In *Proceedings of the 4th workshop on treebanks and linguistic theories (TLT)* (pp. 149–160).
- Quirk, R., Greenbaum, S., & Leech, G., Svartvik, J. (1985). *A comprehensive grammar of the English language*. London: Longman.
- Ramasamy, L., & Žabokrtský, Z. (2012). Prague dependency style treebank for Tamil. In *Proceedings of LREC 2012*. İstanbul, Turkey.
- Rasooli, M. S., Moloodi, A., Kouhestani, M., & Minaei-Bidgoli, B. (2011). A syntactic valency lexicon for Persian verbs: The first steps towards Persian dependency treebank. In *5th language and technology conference (LTC): Human language technologies as a challenge for computer science and linguistics* (pp. 227–231). Poland: Poznań.
- Schwartz, R., Abend, O., & Rappoport, A. (2012). Learnability-based syntactic annotation design. In *Proceedings of COLING 2012: Technical papers* (pp. 2405–2422). India: Mumbai.
- Seginer, Y. (2007). Learning syntactic structure. Ph.D. thesis, University of Amsterdam.
- Simov, K., & Osenova, P. (2005). Extending the annotation of BulTreeBank: Phase 2. In *The fourth workshop on treebanks and linguistic theories (TLT 2005), Barcelona* (pp. 173–184).

- Smrž, O., Bielický, V., Kouřilová, I., Kráčmar, J., Hajič, J., & Zemánek, P. (2008). Prague Arabic dependency treebank: A word on the million words. In *Proceedings of the workshop on Arabic and local languages (LREC 2008)* (pp. 16–23). Marrakech, Morocco. European Language Resources Association.
- Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., & Nivre, J. (2008). The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of CoNLL*.
- Taulé, M., Martí, M.A., & Recasens, M. (2008). AnCora: Multilevel annotated corpora for Catalan and Spanish. In *LREC*. European Language Resources Association.
- Tesnière, L. (1959). *Éléments de syntaxe structurale*. Paris: Klincksieck.
- Tsarfaty, R., Nivre, J., & Andersson, E. (2011). Evaluating dependency parsing: Robust and heuristics-free cross-annotation evaluation. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 385–396). Edinburgh, Scotland, UK. Association for Computational Linguistics.
- van der Beek, L., Bouma, G., Daciuk, J., Gaustad, T., Malouf, R., van Noord, G., Prins, R., & Villada, B. (2002). Chapter 5. The Alpino dependency treebank. In *Algorithms for linguistic processing NWO PIONIER progress report*. Groningen, The Netherlands.
- Zeman, D. (2008). Reusable tagset conversion using tagset drivers. In N. Calzolari, K. Choukri, B. Maegaard, Mariani J., J. Odijk, S. Piperidis, & D. Tapias (Eds.), *Proceedings of the sixth international language resources and evaluation conference, LREC 2008* (pp. 28–30). Marrakech, Morocco. European Language Resources Association (ELRA).
- Zeman, D., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtský, Z., & Hajič, J. (2012). HamleDT: To parse or not to parse? In N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the eighth international conference on language resources and evaluation (LREC'12)*. Istanbul, Turkey. European Language Resources Association (ELRA).

6.4 Universal Dependencies

Full reference: Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, 2021. DOI 10.1162/COLI_a_00402. URL <https://aclanthology.org/2021.c1-2.11.pdf>. [de Marneffe et al., 2021]

Comments: The story of Universal Dependencies (Section 3.3) is atypical. Many projects are first publicized in a paper, then the impact of the publication is observed and eventually new work and new papers emerge. In the case of UD, the impact of the project was already well observable when the first descriptive paper appeared at LREC 2016 [Nivre et al., 2016]. The paper described version 1 of the annotation guidelines but later that year we projected the initial experience to version 2, which is still in use today. A paper describing version 2 was published at LREC 2020 [Nivre et al., 2020]. However, here I wish to emphasize and include the article we published a year later in *Computational Linguistics*. In comparison to the LREC papers it puts less weight on the growth and coverage of the data collection and focuses more on the linguistic theory behind the UD framework, which it lays out in much finer detail, with numerous examples from typologically diverse languages. Besides, I can claim significantly larger share of authorship of the latter article. My contribution: 25%. Number of citations according to Google Scholar (retrieved 2023-07-21): **278**, together with the other two papers: **2113**.

Besides working on the UD annotation scheme, I have also converted, annotated or contributed to dozens of UD treebanks. A few of these contributions were described in separate papers:

- Catalan and Spanish [Martínez Alonso and Zeman, 2016]
- Russian [Lyashevskaya et al., 2016, Droганova et al., 2018]
- Arabic [Taji et al., 2017]
- Slovak [Zeman, 2017]
- Latin [Cecchini et al., 2018, Gamba and Zeman, 2023]
- Sanskrit [Dwivedi and Zeman, 2018]
- Bhojpuri [Ojha and Zeman, 2020]
- Yoruba [Ishola and Zeman, 2020]
- Albanian [Toska et al., 2020]
- Indonesian [Alfina et al., 2020]
- Malayalam [Stephen and Zeman, 2023]

Universal Dependencies

Marie-Catherine de Marneffe

The Ohio State University
Department of Linguistics
demarneffe.1@osu.edu

Christopher D. Manning

Stanford University
Department of Linguistics
manning@cs.stanford.edu

Joakim Nivre

Uppsala University
Department of Linguistics and Philology
joakim.nivre@lingfil.uu.se

Daniel Zeman

Charles University
Faculty of Mathematics and Physics
zeman@ufal.mff.cuni.cz

Universal dependencies (UD) is a framework for morphosyntactic annotation of human language, which to date has been used to create treebanks for more than 100 languages. In this article, we outline the linguistic theory of the UD framework, which draws on a long tradition of typologically oriented grammatical theories. Grammatical relations between words are centrally used to explain how predicate–argument structures are encoded morphosyntactically in different languages while morphological features and part-of-speech classes give the properties of words. We argue that this theory is a good basis for crosslinguistically consistent annotation of typologically diverse languages in a way that supports computational natural language understanding as well as broader linguistic studies.

1. Introduction

Universal dependencies (UD) is at the same time a framework for crosslinguistically consistent morphosyntactic annotation, an open community effort to create morphosyntactically annotated corpora for many languages, and a steadily growing collection of such corpora. In all these respects, UD has undeniably been very successful, growing in only six years from ten treebanks and a dozen researchers to 183 treebanks for 104

Submission received: 2 July 2020; revised version received: 9 February 2021; accepted for publication: 25 February 2021.

https://doi.org/10.1162/COLI_a_00402

© 2021 Association for Computational Linguistics
Published under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license

languages with contributions from 416 researchers around the world.¹ UD treebanks are now widely used in natural language processing research, including but not limited to research on syntactic and semantic parsing, and increasingly also in linguistic research, particularly on psycholinguistics and word order typology.

Some people think that UD is only a tool for annotation, and as such a rather eclectic approach building on existing de facto standards with many practical compromises. Although UD borrows terminology and concepts from many earlier grammatical theories, it is nevertheless a coherent theory resulting from a large amount of careful community work aiming at a principled but broadly applicable view of morphology and syntax. We believe that a clearer description of the underlying theory will help people to fully understand UD, its merits, and its limitations, and we attempt to articulate that theory, in particular, for version 2 of UD, in this article.²

The article is organized as follows. Section 2 introduces the basic theoretical assumptions of UD, including a commitment to words and grammatical relations as fundamental building blocks of grammatical structure. Section 3 is a survey of linguistic constructions and their analysis in UD, with examples from a broad range of languages. In Section 4, we zoom in on core arguments, which play a central role in UD, and discuss how they can be analyzed across typologically different languages. Section 5 discusses the design principles of UD against the backdrop of previous sections and Section 6 concludes with a brief outlook.

2. Basic Tenets of UD

The goal of UD is to offer a linguistic representation that is useful for morphosyntactic research, semantic interpretation, and for practical natural language processing across different human languages. It therefore puts an emphasis on simple surface representations that allow parallelism between similar constructions across different languages, despite differences of word order, morphology, and the presence or absence of function words.

2.1 Linguistic Representation and Information Packaging

When humans observe the world, they see entities (or objects) that participate in events (actions and states). The organization of all human languages reflects this basic world view. Therefore, in UD, we organize description around the two fundamental linguistic units of a **nominal**, canonically used for representing an entity, and a **clause**, canonically used for representing an event. Both nominals and clauses are often refined by describing an attribute of the entity or event, which can be done by the third fundamental linguistic unit of a **modifier**.

2.1.1 Heads and Dependents. A clause has a main predicate that expresses the state or action, and in most cases, states and actions involve participants expressed as nominals. In such a way, language has a hierarchical structure: Clauses can contain nominals, modifiers, and other clauses; nominals can also contain all three phrasal units; and modifiers

¹ Release v2.7, November 15, 2020. For more information, see <https://universaldependencies.org>.

² Because our focus in this article is theoretical, we do not go into practical matters concerning annotation, treebanks, and parsing. We also do not discuss the historical development of UD. For these aspects we refer to the papers on UD v1 (Nivre et al. 2016) and v2 (Nivre et al. 2020) and to the UD Web site (<https://universaldependencies.org>).

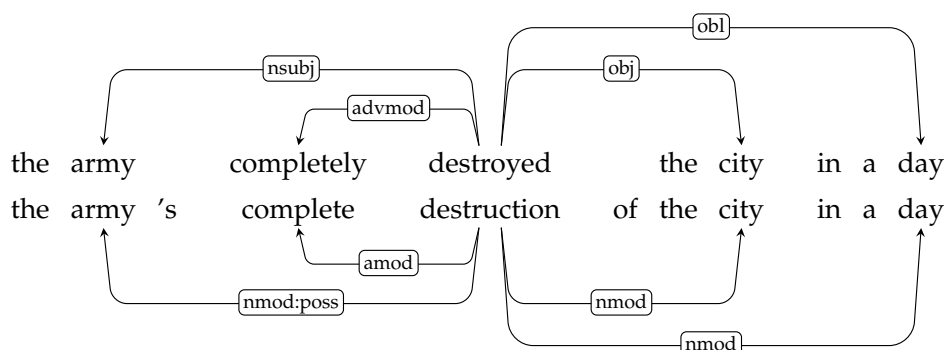


Figure 1
Partial UD analysis for a clause (top) and a nominal (bottom).

can contain modifiers. To express these ideas in UD, we adopt a **dependency grammar** perspective: A phrase has a **head** and other things that it contains are **dependents** of that head.

Dependency is a binary asymmetrical relation, which we represent in diagrams by an arrow from the head to the dependent (or, more precisely, to the head word of the dependent, when the dependent is itself a multiword unit), as in Figure 1. Through these dependencies, the words of a sentence are organized into a tree structure with the main predicate as the root.³ Dependencies are typed with grammatical relation labels, as further discussed in Section 2.3. The head in a dependency is informally the main word of a phrase. The head of a nominal is canonically a noun. The head of a clause, commonly referred to as the **predicate**, is most commonly a verb but may also be an adjective or adverb, or even a nominal. The most common modifier heads are adjectives and adverbs. Sometimes linguistic head functions are divided between a structural center (an auxiliary or function word) and a semantic center (a lexical or content word), such as for periphrastic verb tenses like *has arrived*. This is what Tesnière (2015 [1959], ch. 23) refers to as a **dissociated nucleus**. In such cases, UD chooses the lexical or content word as the head, and makes function words dependents of the head in the dependency tree structure, while recognizing that they do form a nucleus together with the content word. A consequence of this decision, further discussed in Section 2.3.3, is that a UD tree represents a sentence's observed surface predicate–argument structure rather than necessarily accurately capturing phrase-internal syntactic constituency.

2.1.2 Nominals, Clauses, and Modifiers. In more detail, UD assumes a simple typology of three kinds of phrasal units (which might minimally be just a single word):

1. Nominals: the primary means for referring to entities
2. Clauses: the primary means for referring to events
3. Modifiers: the canonical attributive modifiers of nominals, clauses, and other modifiers

³ The tree constraint holds for the *basic* UD representation, which is the focus of this article. UD also defines an *enhanced* representation, which makes explicit additional implicit relations between words (such as propagating relations between conjuncts and adding subject relations for control and raising constructions). For more information about the enhanced representation, which is a rooted directed graph, see Nivre et al. (2020).

Nominals are similar to the notion of a noun phrase or determiner phrase in many theories, but encompass the entire nominal extended projection (Grimshaw 1991 [2005]), also covering prepositional phrases. While the basic use of nominals is always to refer to entities, they may be used in other functions. For example, most languages allow the nominalization of an event: *The continuation of hostilities* describes an event, but has the syntactic form of a nominal. Clauses can be either the root sentence or an embedded clause, typically express events and states, and have a main predicate, which is canonically a verb but can be other parts of speech used predicatively.

Both nominals and clauses can have their meaning added to by the presence of modifying phrases. Sometimes these phrases are themselves nominals or clauses. For example, in Example (1a), there is a nominal modifying a clause; in Example (1c), there is a nominal modifying a larger nominal; and in Example (1d), there is a clause modifying a nominal. However, sometimes modifying phrases are single words or smaller modifying phrases that do not expand into the same rich structures as nominals and clauses. We describe this third class of linguistic units as **modifiers**. In Example (1b), there is a modifier modifying a clause, and in Example (1e), there is a modifier modifying a nominal. Modifiers can themselves be modified: The modifier *somewhat* modifies *rusty* in Example (1e). It is generally true in languages that there is not an infinite regress: The modifiers of modifiers are limited and normally of the form of basic modifiers, and so we continue to call them all modifiers.

- (1) a. [He opened the can [with a screwdriver]]
 b. [He opened the can [carefully]]
 c. [the screwdriver [on the table]]
 d. [the screwdriver [which my mother bought me]]
 e. [the [[somewhat] rusty] screwdriver]

This taxonomy is not unique to UD. As it reflects the basic structure of human language, similar taxonomies can be found in many other frameworks, especially those starting from a functional or typological perspective on language. For example, Croft (1991, forthcoming) distinguishes **reference**, **predication**, and **modification** as three basic information packaging functions, or propositional act functions, underlying syntactic constructions. These correspond straightforwardly to the canonical usages of our nominals, clauses, and modifiers, respectively.

The distinction between nominals and clauses is fundamental to UD, which systematically uses different dependency relations in the two types of structures, as illustrated in Figure 1. The clause *the army completely destroyed the city in a day* is headed by the verbal predicate *destroyed*, while the nominal *the army's complete destruction of the city in a day* is headed by the noun *destruction*. The predicate has two core arguments (*the army*, *the city*), while the noun has two genitive modifiers accompanied by different kinds of case markers (*the army's*, *of the city*). The adverbial modifier (advmod, *completely*) of the predicate corresponds to an adjectival modifier (amod, *complete*) of the noun. Even the temporal modifier *in a day*, which has the form of a prepositional phrase in both cases, is classified as an oblique modifier (obl) of the predicate but as a nominal modifier (nmod) of the noun. Similarly, the typology of dependency relations also captures whether the dependent is a nominal, a clause, or a modifier. For example, a modifier of a nominal will be respectively a nominal modifier (nmod), an adjectival modifier (amod), or an adnominal clause (acl) depending on the type of the dependent. Hence phrasal types are recoverable without being explicitly represented.

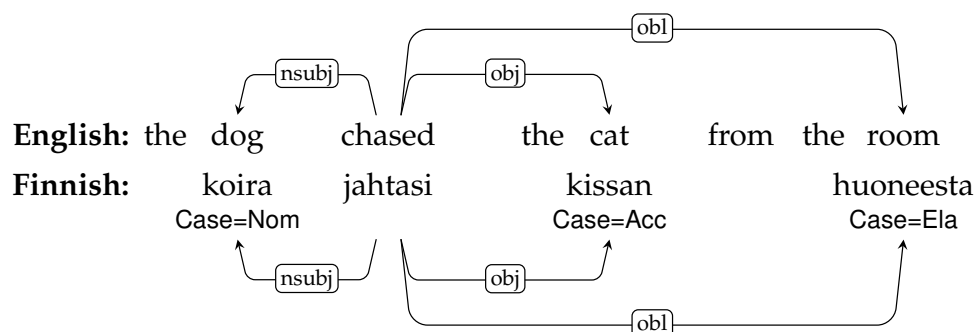


Figure 2
Simplified UD annotation for equivalent sentences from English (top) and Finnish (bottom).

2.2 Words as Basic Units

UD follows traditional grammar in giving primary status to **words**. Words are the basic elements connected by dependency relations; they have morphological properties and enter into syntactic relations. The primacy of words can be understood as a commitment to the lexical integrity principle (Chomsky 1970; Bresnan and Mchombo 1995; Aronoff 2007), which states that words are built out of different structural elements and by different principles of composition than syntactic constructions. Despite the challenges in defining words in a crosslinguistically consistent manner—faced with phenomena like clitics, compounding, and incorporation, to mention only a few⁴—we believe that this approach is more interpretable and useful for most potential users of UD and generalizes better across languages than trying to segment words into smaller units like morphemes. This view is further supported by developments in morphological theory, which favor word-based abstractive models over morpheme-based constructive models (Stump 2001; Blevins 2006; Blevins, Ackerman, and Malouf 2017).

It is important to note, however, that the morphosyntactic notion of word does not always coincide with orthographical or phonological units. For instance, clitics (Spencer and Luís 2012) often need to be separated from their hosts and treated as independent words even if they are not recognized as such in conventional orthography (for instance, the English *'s* genitive, as in *the army's* in Figure 1, acts as a phrasal clitic, as can be seen by expansions such as *the army of the undead's*). Similarly, compound words need a special treatment, because in some languages their written form may contain boundary markers such as whitespace (as in *night school* in English) whereas in other languages they do not (as in *Abendschule* ‘night school’ in German).

2.2.1 Content Words, Function Words, and Grammaticalization. We expect the words that enter into the main syntactic relations to be autosemantic, that is, content words with an independent meaning—typically verbs, nouns, adjectives, or adverbs, as well as corresponding pro-forms with a contextually determined referential meaning. For instance, the backbone of the UD morphosyntactic representations for the English and Finnish sentences in Figure 2 consists of the three relations that are common to these sentences: argument and modifier relations involving predicates and nominals.

These content words often occur together with grammatical markers, senseless elements that further specify their meaning or syntactic role. Typical examples are

⁴ See also Haspelmath (2011a).

markers of tense, mood, and aspect for (verbal) predicates and of number, definiteness, and case for nominals. As explained in Section 2.1.1, UD attaches such elements as dependents of the content word, analyzing them as parts of dissociated nuclei.

The distinctions between content words and function words, and between function words, clitics, and inflectional morphemes, are not always clear-cut. We know from the literature on grammaticalization that grammatical markers normally develop out of content words and first appear as separate function words but often later become clitics and eventually inflectional affixes, a process sometimes referred to as the cline of grammaticalization (Hopper and Traugott 2003). At any given historical stage, a language will contain constructions that are at intermediate stages of this development, and where it is therefore not straightforward to classify the components of the construction. Consider, for example, the Swedish sentences in Example (2).

- (2) a. Hon kunde (*att) sjunga
 she could to sing
 ‘She could sing’
- b. Hon började (att) sjunga
 she began to sing
 ‘She began to sing’
- c. Hon gillade *(att) sjunga
 she liked to sing
 ‘She liked to sing’

In Example (2a), it is impossible to insert the infinitive marker *att* before the verb *sjunga* ‘sing’, which shows that *kunde* ‘could’ is an auxiliary verb. In Example (2c), it is equally impossible to *omit* the infinitive marker, which shows that *gillade* ‘liked’ is a main verb taking an infinitive complement. In Example (2b), however, the infinitive marker is optional, which makes the status of *började* ‘began’ unclear, all the more as its meaning is mainly aspectual and of a kind that could undergo grammaticalization. In annotation, we are forced to make a somewhat arbitrary choice and make Example (2b) parallel to either Example (2a) or Example (2c)—but not both. Note that in Example (2a), the verb *sjunga* is the head of the sentence, while in Example (2c), *gillade* is the head, so changing the analysis of Example (2b), as grammaticalization proceeds, requires not just a part-of-speech change but a fundamental syntactic reanalysis of the sentence, or what Gerdes and Kahane (2016) refer to as a “catastrophe.” Making particular categorical decisions in such intermediate cases will inevitably add some distortion to our representation of the linguistic reality, but we can only do our best to maintain consistency in these decisions and carefully document the criteria.

Similar issues arise in word segmentation, where it is sometimes difficult to decide whether a grammatical marker should be treated as an inflectional affix, clitic, or function word, despite extensive discussion of discriminative criteria, such as in Zwicky and Pullum (1983).

2.2.2 Part-of-Speech Categories. All linguistic theories assume that words can be classified by a **word class** or **part of speech** (POS) according to their behavior within the language system. Partly for broad comprehensibility, UD stays fairly close to traditional parts of speech, such as the eight parts of speech commonly recognized for English, but it makes a few finer distinctions, better reflecting modern linguistic typology, and adds some classes for punctuation and other symbols. As a result, UD distinguishes 17 coarse-grained classes of words and other elements of text, and assigns them the

Table 1

Universal part-of-speech tags (UPOS). Typos and abbreviations are given the category of the unabbreviated or correct word.

Traditional POS	UPOS	Category
noun	NOUN	common noun
	PROPN	proper noun
verb	VERB	main verb
	AUX	auxiliary verb or other tense, aspect, or mood particle
adjective	ADJ	adjective
	DET	determiner (including article)
	NUM	numeral (cardinal)
adverb	ADV	adverb
pronoun	PRON	pronoun
preposition	ADP	adposition (preposition/postposition)
conjunction	CCONJ	coordinating conjunction
	SCONJ	subordinating conjunction
interjection	INTJ	interjection
–	PART	particle (special single word markers in some languages)
–	X	other (e.g., words in foreign language expressions)
–	SYM	non-punctuation symbol (e.g., a hash (#) or emoji)
–	PUNCT	punctuation

labels (“universal part-of-speech tags,” UPOS) shown in Table 1. These categories are widely attested in the world’s languages. We do not claim that all languages must use all of these categories, but we do assume that every word in every language can be assigned one of them. Some word-class distinctions are particularly important in UD: For example, the dividing line between nouns and verbs plays a significant role in specifying whether a constituent is nominal or clausal (Section 2.1.2).

It is not easy in all cases to define word classes in a crosslinguistically consistent manner. Grammatical criteria used in word classification have to be specific for individual languages, although we do expect similar criteria in languages that are closely related. Because morphological criteria are not sufficient and available for all categories in all languages, in many cases we have to rely primarily on syntactic criteria. For instance, Czech adjectives inflect for three grammatical genders, two numbers, seven cases, and three degrees of comparison. They typically specify properties of nouns and are found right before the nouns they modify. In contrast, only a subset of English adjectives can inflect for degree of comparison, and none inflect for gender, number, or case. Yet their prototypical function and distribution is similar to Czech: If used attributively, they occur right before the nouns whose attributes they specify.

While the definition of word categories is not universal, their names are portable across languages so that same-labeled categories show partially similar syntactic behavior and overlapping semantic content (Schachter and Shopen 2007; Haspelmath 2001; Croft 1991). It is possible to have one category that will contain most words referring to entities, such as *mother*, *dog*, or *house*; words in this category will be called nouns. Similarly, the label “verb” is used for the class of prototypical action words (such as *go*, *buy*, *eat*), and the class of adjectives will likely contain equivalents of *small*, *good*, or *white*. In addition, each of these categories may contain words with less prototypical semantics, if they follow the language-particular rules that define the category. Hence the English nouns include words like *destruction* and *weakness* because their morphological and distributional behavior is noun-like, although their meaning is derived from the verb *destroy* and the adjective *weak*, respectively.

A common difficulty is that words of one category are sometimes used in positions and functions normally associated with a different category, without changing their morphology (if morphological criteria are available at all). For example, an English adjective may appear in the subject position with a definite article but without the modified noun (*the healthy, the sick*). We could treat such examples as instances of ellipsis (where the underlying noun phrase could be *the healthy/sick people*) or we could say that the adjective has been converted into a noun in the given sentence. However, the part-of-speech classification is most useful if it captures regular, prevailing syntactic behavior and does not reflect sentence-specific exceptional behavior. If the POS category were completely predictable from the syntactic function (which is an independent part of UD annotation), then the POS tag would be uninformative. It would also be harder to find interesting crosslinguistic differences, for example, that language X allows words of category A to have syntactic function B, but language Y does not. Therefore in English we assign the ADJ tag to *healthy* even if it heads a nominal phrase.

Sometimes a functional shift is better explained by grammaticalization (see Section 2.2.1) rather than by exceptional usage in a specific sentence. The English adverb *so* is used as an adverb in Example (3a) and Example (3b), but as a discourse connective similar to a coordinating conjunction in Example (3c). However, we keep the word *so* in the adverb category in these three examples.

- (3) a. People work so hard
 b. If you have not done so already
 c. We are aiming to have it next week, so I need to know if you can ship it quickly

Nevertheless, there are situations where we consider the two competing functions too distant and mutually incompatible, and we treat the word as a homonym whose category has to be disambiguated by context. Consider the Spanish examples in Example (4).

- (4) a. los siete candidatos que compiten mañana
 the seven candidates that compete tomorrow
 'the seven candidates that will compete tomorrow'
 b. Descubrimos que los tres reyes estaban aquí
 discovered.1PL that the three kings were here
 'We discovered that the three kings were here'

In Example (4a), the word *que* 'that' is a relative pronoun that represents the subject of the subordinate clause; we tag it PROM. On the other hand, the same word in Example (4b) has no argument role in the subordinate clause; it is merely a marker of subordination. We tag it SCONJ. The same holds for the English word *that* in the corresponding English sentences. Sometimes morphology in a paradigm makes the analysis clear: When English nouns are used as verbs like in *You should butter your bread*, we regard the word as a verb because it participates in a paradigm with usual verb morphology in the past tense or with third singular subject agreement.

2.2.3 Morphological Features. Many classes of words in many languages participate in paradigms of forms that express extra features, such as number or tense. We can further divide the appropriate POS classes into subclasses according to features that express

Table 2
Universal morphological features.

	Feature	Values
pronominal type	PronType	Art Dem Emp Exc Ind Int Neg Prs Rcp Rel Tot
numeral type	NumType	Card Dist Frac Mult Ord Range Sets
possessive	Poss	Yes
reflexive	Reflex	Yes
foreign word	Foreign	Yes
abbreviation	Abbr	Yes
wrong spelling	Typo	Yes
gender	Gender	Com Fem Masc Neut
animacy	Animacy	Anim Hum Inan Nhum
noun class	NounClass	Bantu1-23 Wol1-12 . . .
number	Number	Coll Count Dual Grpa Grpl Inv Pauc Plur Ptan Sing Tri
case	Case	Abs Acc Erg Nom Abe Ben Cau Cmp Cns Com Dat Dis Equ Gen Ins Par Tem Tra Voc Abl Add Ade All Del Ela Ess Ill Ine Lat Loc Per Sub Sup Ter
definiteness	Definite	Com Cons Def Ind Spec
comparison	Degree	Abs Cmp Equ Pos Sup
verbal form	VerbForm	Conv Fin Gdv Ger Inf Part Sup Vnoun
mood	Mood	Adm Cnd Des Imp Ind Irr Jus Nec Opt Pot Prp Qot Sub
tense	Tense	Fut Imp Nfut Past Pqp Pres
aspect	Aspect	Hab Imp Iter Perf Prog Prosp
voice	Voice	Act Antip Bfoc Cau Dir Inv Lfoc Mid Pass Rcp
evidentiality	Evident	Fh Nfh
polarity	Polarity	Neg Pos
person	Person	0 1 2 3 4
politeness	Polite	Elev Form Humb Infm
clusivity	Clusivity	In Ex

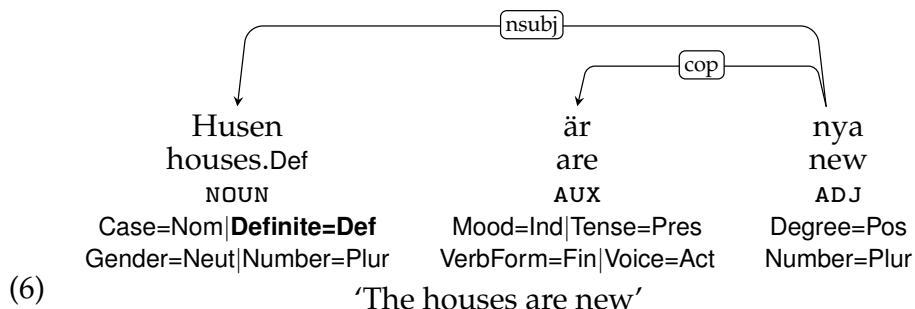
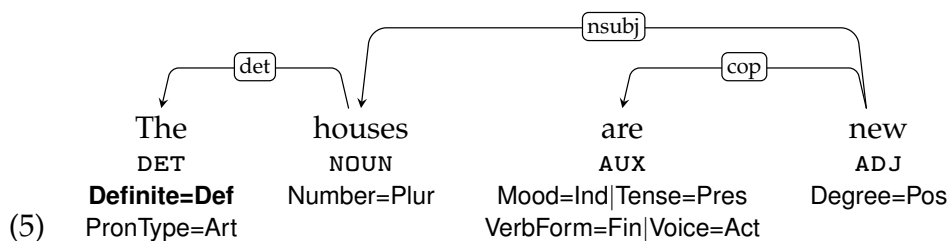
paradigmatic position. For example, the `VerbForm` feature distinguishes the finite verb from various nonfinite forms, which often show a mix of verbal properties and properties of other classes (nouns, adjectives, or adverbs). Depending on the language, it is then possible to distinguish between a finite verb (`VERB VerbForm=Fin`), a verbal participle (`VERB VerbForm=Part`), a deverbal participial adjective (`ADJ VerbForm=Part`), and a common adjective (`ADJ`). It seems quite possible to define a universal set of features, covering what is described by morphology in the world’s languages, and our system in UD is in line with other attempts to do this, such as UniMorph (Sylak-Glassman et al. 2015) and the GOLD Ontology (Farrar and Langendoen 2003).

UD defines a set of feature-value pairs that are attested in multiple languages (Table 2).⁵ Additional features may be defined in language-specific documentation for use in individual languages. Some features are lexical, meaning that the same value of the feature applies to the entire paradigm, that is, to all forms that share the same lemma. Such features serve to further partition the space of word categories by providing distinctions that are more fine-grained, or that cut across the boundaries of the

⁵ In the examples throughout this article, we show only selected Feature=Value pairs that we think are useful to understand the example. The actual UD annotation may contain other features that we omit in the interest of space.

main UPOS categories. Prominent examples are PronType and NumType. For example, the interrogative and indefinite pronominal types are recognized with pronouns (*who* vs. *somebody*), determiners (*which* vs. *some*), as well as with adverbs (*where* vs. *somewhere*). Other features are inflectional, namely, different forms in a word's paradigm may have different values of the feature. A typical example, attested in many languages, is Number: A noun may have a special form when referring to more than one entity, a verb may cross-reference an argument and signal that it is a group of entities, and in many languages the nominal and the verbal inflection coexist, possibly accompanied by number inflection of other categories, such as adjectives. Finally, there are features whose nature differs depending on the part-of-speech category. For instance, Gender is a lexical feature of nouns but it is also an inflectional feature of words that show morphological agreement with nouns, such as verbs or adjectives.

A feature may also be marked on a function word that contributes the feature to a dissociated nucleus. For example, definiteness of nouns is expressed morphologically in Swedish (*husen* 'the houses' vs. *hus* 'house(s)'), hence Definite is an inflectional feature of Swedish nouns, but English nominals derive their definiteness from definite or indefinite articles (as shown in Example (5) and Example (6)). These are function words and the definite article is assigned a different lemma than the indefinite article, hence Definite is a lexical feature of articles (determiners) in English.



UD does not impose any universal constraints on compatibility between features and part-of-speech categories. Any feature value can potentially occur with any UPOS tag. However, constraints of this sort typically exist at the language-particular level. Hence, for instance, the Number feature is defined for English nouns and verbs but not adjectives, while in Czech it is defined for adjectives, too. The Case feature in English appears marginally with certain pronouns and uses only two values, while in Czech it has seven values and is defined for nouns, adjectives, pronouns, determiners, and numerals.

In some languages, some features are marked more than once on the same word. We say that there are several **layers** of the feature. The exact meaning of individual layers is language-dependent. For example, possessive adjectives, determiners, and pronouns may have two different values of both Gender and Number. One of the values is determined by agreement with the modified (possessed) noun. This is parallel to other (non-possessive) adjectives and determiners that agree in gender and number with the nouns they modify. The other value is determined lexically because it is a property of

the possessor. Layers are indicated by their identifier in square brackets after the feature name. For example, the Czech possessive pronoun *náš* ‘our’ is tagged `Number[psor]=Plur` to indicate that the possessor’s number is plural. At the same time, the word form refers to a singular possessee; this layer of the Number feature is considered default for Czech possessives and needs no layer identifier: `Number=Sing`.

Where necessary, UD allows language-specific features, for example, `FocusType` has been used in the Niger–Congo language Wolof but it has not yet been established as applicable in other languages. It is used with Wolof auxiliaries, which, among other things, indicate whether the focus is on the subject of the clause, the verb, or the verb’s complement (Dione 2019).

2.3 Grammatical Relations between Words

Perhaps the most distinctive feature of UD is its taxonomy of grammatical relations between words.⁶ Each dependent of a head, and also any function words that belong with a head, are connected to the head via a grammatical relation drawn from a universal typology of 37 grammatical relations, listed in Table 3. As discussed earlier, the grammatical relations are organized around whether the head is the head of a clause or nominal, and whether the dependent is a clause, nominal, or modifier, although a number of other distinctions and special cases, prominent in the world’s languages, are also represented. Table 4 illustrates the organization of the grammatical relations. The root relation is used for the root of the sentence, with a dummy head that does not need to be explicit. The `dep` relation is used when no other relations are deemed appropriate. The relations are illustrated throughout Section 3. The set of allowed relations is closed, but UD allows relation subtypes separated from the main relation by a colon to provide further distinctions or to capture language-specific constructions. For example, a number of languages mark relative clauses as `acl:relcl` and predeterminers as `det:predet`.

2.3.1 Usefulness of Grammatical Relations for Linguistic Typology. One of the basic tenets of UD is that *grammatical relations* like *subject* and *object* provide a useful level of abstraction to account for the complex mapping from overt coding properties like case-marking, agreement, and word order to the underlying semantic predicate–argument structure of sentences. In particular, they provide a happy middle ground of usually being easily surface-form identifiable while being useful for crosslinguistic description.

In this respect, UD follows in the tradition of theories as diverse as relational grammar (Perlmutter 1983), lexical-functional grammar (LFG) (Kaplan and Bresnan 1982; Dalrymple 2001; Bresnan et al. 2016), word grammar (Hudson 1984, 1990), functional generative description (FGD) (Sgall, Hajičová, and Panevová 1986), meaning-text theory (MTT) (Mel’čuk 1988; Milicevic 2006), role and reference grammar (Van Valin, Jr. 1993), and head-driven phrase structure grammar (Pollard and Sag 1994). Moreover, grammatical relations have always played a prominent role in linguistic typology, starting with the pioneering works of Greenberg (1963) and then Comrie (1981), and continuing in contemporary work like that of Croft (2001, 2002), Andrews (2007), Dixon (2009), and Haspelmath (2011b). Although the universality of grammatical relations is sometimes debated, their status as useful theoretical constructs for crosslinguistic studies is rarely questioned.

⁶ We generally use the term *grammatical relation* rather than *grammatical function* or *dependency label*, but we regard the terms as essentially synonymous—unlike, for example, Andrews (2007).

Table 3

The 37 syntactic relations in UD, with a brief explanation of the relation and a reference to an example.

Relation	Definition	Ex.
acl	adnominal clause; finite or non-finite clause modifying a nominal	(28)
advcl	adverbial clause modifying a predicate or modifier word	(27)
advmod	adverb or adverbial phrase modifying a predicate or modifier word	(20a)
amod	adjectival modifier of a nominal	(12)
appos	appositional modifier; a nominal used to define, name, or describe the referent of a preceding nominal	(15)
aux	auxiliary; links a function word expressing tense, mood, aspect, voice, or evidentiality to a predicate	(16c)
case	links a case-marking element (preposition, postposition, or clitic) to a nominal	(9)
cc	links a coordinating conjunction to the following conjunct	(23)
ccomp	clausal complement of a verb or adjective without an obligatorily controlled subject	(26b)
clf	(numeral) classifier; a word reflecting a conceptual classification of nouns linked to a numeric modifier or determiner	(11)
compound	any kind of word-level compounding (noun compound, serial verb, phrasal verb)	(37)
conj	conjunct; links two elements which are conjoined	(23)
cop	copula; links a function word used to connect a subject and a nonverbal predicate to the nonverbal predicate	(17a)
csubj	clausal syntactic subject of a predicate	(25)
dep	unspecified dependency, used when a more precise relation cannot be determined	
det	determiner (article, demonstrative, etc.) in a nominal	(10)
discourse	discourse element (interjection, filler, or non-adverbial discourse marker)	(20b)
dislocated	a peripheral (initial or final) nominal in a clause that does not fill a regular role of the predicate but has roles such as topic or afterthought	(22b)
expl	expletive; links a pronominal form in a core argument position but not assigned any semantic role to a predicate	(22c)
fixed	fixed multiword expression; links elements of grammaticalized expressions that behave as function words or short adverbials	(39)
flat	flat multiword expression; links elements of headless semi-fixed multiword expressions like names	(40)
goeswith	links parts of a word that are separated but should go together according to standard orthography or linguistic wordhood	(44)
iobj	indirect object; nominal core argument of a verb that is not its subject or (direct) object	(16c)
list	links elements of comparable items interpreted as a list	(46)
mark	marker; links a function word marking a clause as subordinate to the predicate of the clause	(27a)
nmod	nominal modifier; a nominal modifying another nominal	(13)
nummod	numeric modifier; numeral in a nominal	(10)
nsubj	nominal subject; nominal core argument which is the syntactic subject (or pivot) of a predicate	(16)
obj	object; the core argument nominal which is the most basic core argument that is not the subject, typically the most directly affected participant	(16)
obl	oblique; a nominal functioning as a non-core (oblique) modifier of a predicate	(21)
orphan	links orphaned dependents of an elided predicate	(43)
parataxis	links constituents placed side by side with no explicit coordination or subordination	(32)
punct	punctuation attached to the head of its clause or phrase	(23b)
reparandum	repair of a (normally spoken language) disfluency	(45)
root	root of the sentence	(16)
vocative	nominal directed to an addressee	(22a)
xcomp	clausal complement of a verb or adjective with an obligatorily controlled subject	(26a)

2.3.2 Core Arguments and Oblique Modifiers. In classifying grammatical relations, UD distinguishes the **core arguments** of a predicate, essentially subjects and objects, from all other dependents at the clause level, collectively referred to as **oblique modifiers**. The core–oblique distinction is commonly assumed in typological linguistics (see, e.g., Thompson 1997; Andrews 2007) and is ultimately an information packaging distinction.

Table 4
Typology of the syntactic relations.

Head \ Dependent	Nominals	Clauses	Modifier words	Function words
Clausal core arguments	nsubj obj iobj	csubj ccomp xcomp		
Clausal non-core arguments	obl vocative expl dislocated	advcl	advmod discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	MWE	Loose	Special	Other
conj	fixed	list	orphan	punct
cc	flat compound	parataxis	goeswith reparandum	root dep

All or nearly all languages have a standard way of encoding the one or two arguments of most verbs, and this unmarked form of argument expression defines core arguments for that language. The specific criteria used to identify core arguments are ultimately language-specific, but the following criteria recur in many languages:

- Verbs usually only agree with core arguments.
- Core arguments often appear as bare nominals while obliques are marked by adpositions or other grammatical markers.
- Core arguments often appear in certain cases, traditionally called nominative, accusative, and absolutive.⁷
- Core arguments in many languages occupy special positions in the clause, often adjacent to the verb.
- Properties such as being the controller of a subordinate clause argument are often limited to core arguments.
- Valency-changing operations such as passive, causative, and applicative are often restricted to the promotion or demotion of core arguments.

UD assumes that all languages have a way of identifying usually two core arguments, and reserves the relations of **subject** and **object** for these. If additional dependents that are treated similarly to the basic core arguments appear in a clause, with or without valency-changing operations targeting them, these are also regarded as core arguments. For example, some languages allow indirect (or secondary) objects, while other languages do not.

It is important to note that status as a core argument is decoupled from the semantic role of a participant. Depending on the meaning of a verb, many different semantic

⁷ See Section 4.4 for a discussion of ergative case.

roles can be expressed by the same means of encoding core arguments. Nevertheless, there is a correlation: Agent and patient or theme roles of predicates in their unmarked valence are normally realized as core arguments. It is also important to note that the core–oblique distinction has to do with the morphosyntactic encoding of dependents, not with their status as obligatory or selected by the predicate. Thus, UD does not assume the traditional argument–adjunct distinction found in many linguistic theories, which we take to be sufficiently subtle and hard to apply consistently both within and across languages that the best solution is to avoid it. This position has been defended on theoretical grounds by Haspelmath (2014) and Przepiórkowski (2016), and is also adopted for practical reasons in many treebanks, notably the Penn Treebank for English (Marcus, Santorini, and Marcinkiewicz 1993). The distinction between core arguments and oblique modifiers is only applied at the clausal level; all dependents of nominals are treated as oblique.

2.3.3 UD as Tectogrammar. The emphasis on grammatical relations makes UD representations similar to syntactic representations that are midway between surface constituency and argument structure in multistratal theories, such as the *f*-structures in LFG (Bresnan et al. 2016), the deep syntactic or tectogrammatical representations in multistratal versions of dependency grammar (Sgall, Hajičová, and Panevová 1986; Mel'čuk 1988), or final relations in relational grammar. In particular, UD captures the observed surface predicate–argument structure rather than any sort of abstracted or underlying deeper structure. However, being a monostratal theory, UD also needs to incorporate aspects of surface realization, such as word order, function words, and morphological inflections, which typically belong to a separate surface-oriented representation in multistratal theories. As a result, UD representations end up looking like a hybrid of deep and surface-oriented representations, but where the tree structure is primarily determined by predicate–argument structure. We believe the failure to appreciate this to be one of the primary causes for misunderstandings about the theoretical foundations of UD. More specifically, this means that UD represents classic surface constituency only to the level of demarcating clauses, nominals, and modifiers. The internal structure of each of these phrases represents predicates and grammatical relations, somewhat similarly to an LFG *f*-structure, an MTT SyntR, or an FGD tectogrammatical representation, and commonly does not capture fine details of surface constituency as regards auxiliary verbs, adpositions, and so on.⁸

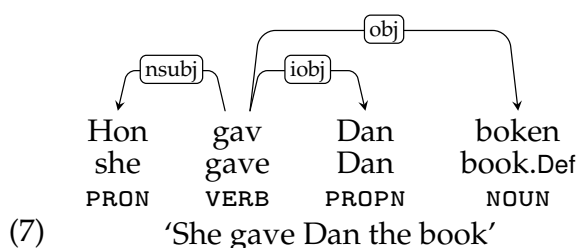
3. Analyzing Linguistic Constructions in UD

Having explained the basic principles of UD as a linguistic theory, we now illustrate how this theory can be applied to a range of linguistic phenomena. We start with nominals and (simple) clauses, as the most fundamental constructions, and gradually move on to more complex phenomena, including some that are ubiquitous in language use but not often discussed in grammars.

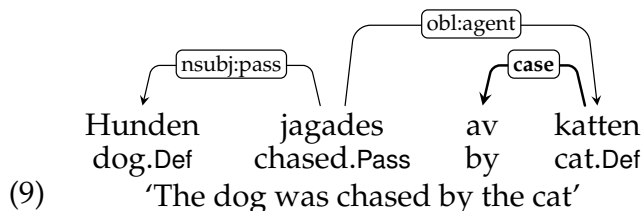
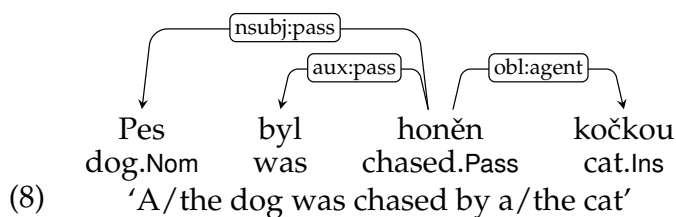
⁸ In contrast, Osborne and Gerdes (2019) and Gerdes et al. (2018) argue for and present a dependency annotation model that does respect surface constituency, while other annotation schemes are closer to (semantic) argument structure (Baker, Fillmore, and Lowe 1998; Palmer, Gildea, and Kingsbury 2005).

3.1 Nominals

Nominals⁹ are a fundamental linguistic unit in all languages, and typically refer to entities (in a wide sense). Nominals occur as core arguments of predicates and in a range of other functions, including predicative uses. In the simplest case, a nominal consists of a single head word, which is typically a noun (NOUN), proper noun (PROPN), or pronoun (PRON). Depending on the language, nominal head words may carry a number of morphological features, of which the most common are gender (Gender), number (Number), case (Case), and definiteness (Definite). In the Swedish Example (7), the subject nominal is the pronoun *hon* ‘she’, the indirect object nominal is the proper noun *Dan*, and the direct object nominal is the noun *boken* ‘the book’.



3.1.1 Case Markers. Case marking is one of the strategies that languages use to encode the grammatical function of a nominal. Case marking can be realized through morphological inflection (captured in UD by the Case feature) or by clitics or adpositions (prepositions and postpositions). In the interest of crosslinguistic parallelism, UD takes a radical approach and treats all adpositions as case markers, attaching them to the nominal head with the special case relation.¹⁰ This allows us to analyze the following examples as both having a direct dependency relation from the predicate to the nominal filling the (oblique) agent role of a passive, despite the fact that Czech Example (8) uses a noun in the instrumental case (*kočkou*) while Swedish Example (9) adds a preposition (*av*):



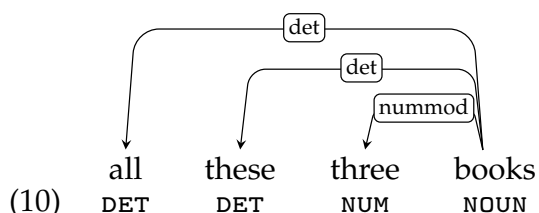
⁹ The term *nominal* is roughly equivalent to the more commonly used term *noun phrase*. However, we prefer the term *nominal* both because phrases are not primitive notions in UD and because we include among nominals some constructions that would not normally be classified as noun phrases, notably, prepositional phrases.

¹⁰ In the typological linguistics literature, Haspelmath (2019) also argues for a unified treatment of case markers and adpositions, suggesting it is “very unclear how they could be distinguished consistently as comparative concepts.”

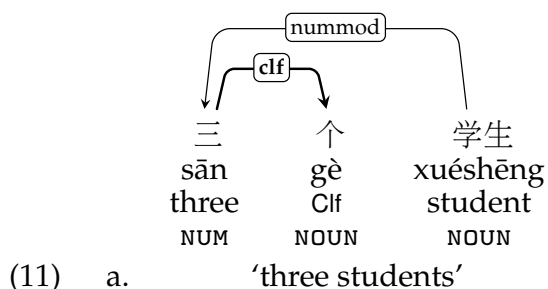
This means that prepositional (and postpositional) phrases are treated in UD as nominals, where the nominal head is the referential core while the adposition is a functional marker. This can be seen as an instantiation of Tesnière’s notion of a dissociated nucleus and does not entail that the adposition is seen as a syntactic dependent of the noun in the narrow sense.

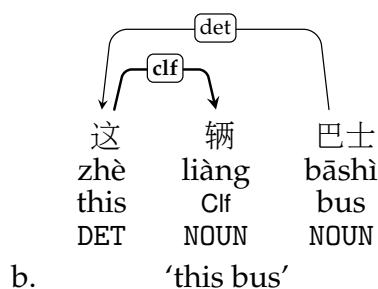
3.1.2 Determiners, Numerals, and Classifiers. Nominals headed by nouns often contain determiners, which can be roughly divided into four classes: articles, demonstratives, interrogatives, and quantifiers. Articles, like English *a(n)* and *the*, specify definiteness or related properties. They are obligatory in some languages (at least with some types of nouns), and completely absent in others. Demonstratives, like Latin *hic* ‘this’, *iste* ‘that (of yours)’ and *ille* ‘that’, anchor the noun phrase deictically and seem to be available in all languages. Interrogatives, like English *which*, are used to form noun phrases that can be used in interrogative (and sometimes relative) clauses. Quantifiers, like French *tout* ‘all’, *quelque* ‘some’, and *aucun* ‘any’, specify quantity or existence of the referent. In many languages, different determiners are in complementary distribution or have special constraints on their cooccurrence and possible order. Regardless of whether a noun phrase contains one or more determiners, UD uses the *det* relation to connect them all directly to the nominal head, as illustrated in Example (10) below.

Nominals headed by nouns may also contain numerals, which express exact numerical quantities (1, 2, 3, ...). Numerals resemble determiners and can often replace them (*one book* vs. *a book* or *this book*) but have special properties in many languages, in particular in relation to classifiers (see below), and UD therefore uses the special *nummod* relation to connect a numeral to the head noun, as in Example (10). Note that the *nummod* relation is only used for cardinal numerals (*one, two, three*). Ordinal numerals (*first, second, third*) are instead treated as adjectives both morphologically and syntactically.



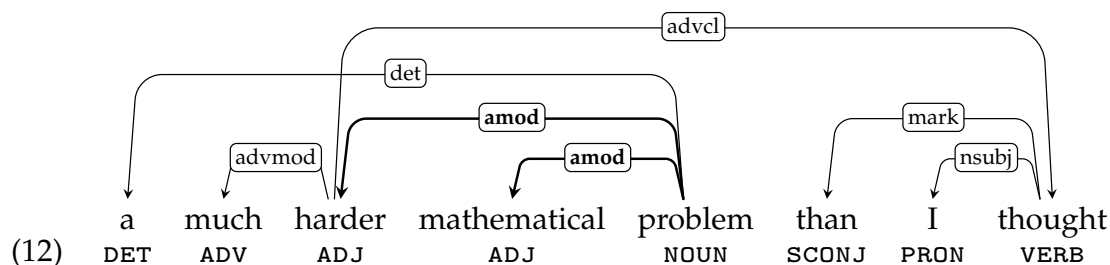
A classifier is a word that accompanies a noun in certain grammatical contexts. The prototypical case is that of numeral classifiers, where the word is used with a numeral for counting objects and where the numeral normally cannot occur without the classifier. A classifier generally reflects some kind of conceptual classification of nouns, based principally on features of their referents. UD uses the *clf* relation to connect the classifier to the numeral (or determiner) together with which it modifies the noun, as in Example (11) from Chinese.



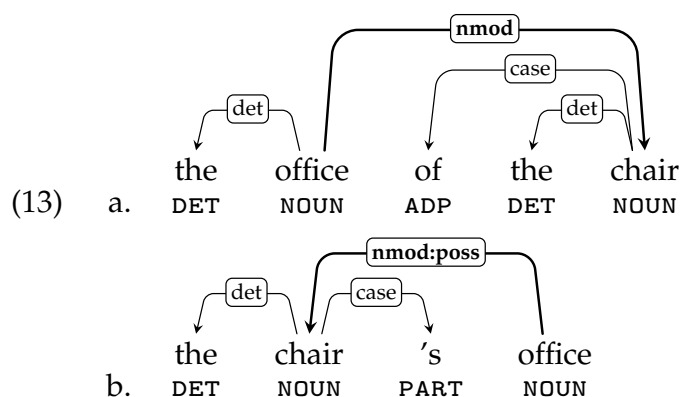


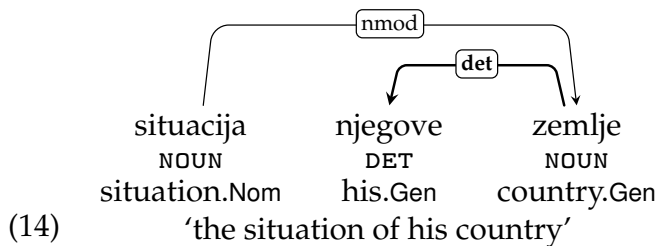
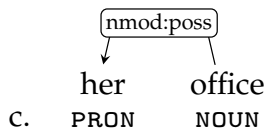
The morphological analysis of classifiers is debated. Etymologically, classifiers are normally nouns, and UD generally recommends using the `NOUN` tag. It has been suggested that a special feature should be added to distinguish the classifier use, since the words can normally also be used as regular nouns, but there is currently no such feature.

3.1.3 Adjectival and Nominal Modifiers. Adjectives modifying the head of a nominal are linked to the head noun with the `amod` relation. Unlike case markers, determiners, numerals, and classifiers, adjectives can be freely multiplied and can themselves be the head of complex constructions involving modifiers of various kinds, as illustrated in Example (12). A special case of adjectival modifiers are ordinal numerals (as in *the second Harry Potter book*), which are analyzed as adjectives in UD.

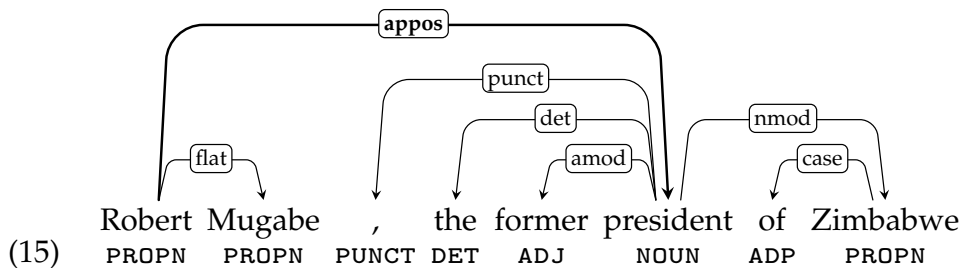


The head of a nominal may also be modified by another nominal, whose head is then linked to the higher noun with the `nmod` relation as in Example (13a). A special case is the genitive construction as in Example (13b), which may occur with or without overt case markers. Possessive pronouns, when used to modify nouns, are treated as a special case of nominal modifiers. In many treebanks, the subtype `nmod:poss` is used both for possessive pronouns Example (13c) and full genitive noun phrases Example (13b). However, if the grammatical rules of the language treat the possessive word analogously to determiners (i.e., the possessive is not a nominal), `det (:poss)` is used as in the Croatian Example (14).





A special type of nominal modification, recognizable in some languages, is apposition, for which UD has a special *appos* relation. It connects two nominals that have the same (or overlapping) referents, as exemplified in Example (15). According to the UD criteria, the two nominals involved in an appositive construction are syntactically independent, can often be reordered, and are usually separated by a comma in writing. The *appos* relation is also strictly left-to-right, meaning that the first nominal is always treated as the head. This is a more narrow-scoped definition than the notion of apposition found in some grammars, which may also include modifiers that precede the head or that are not themselves syntactically independent nominals.

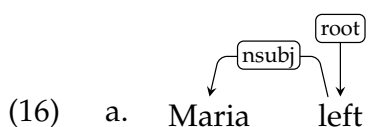


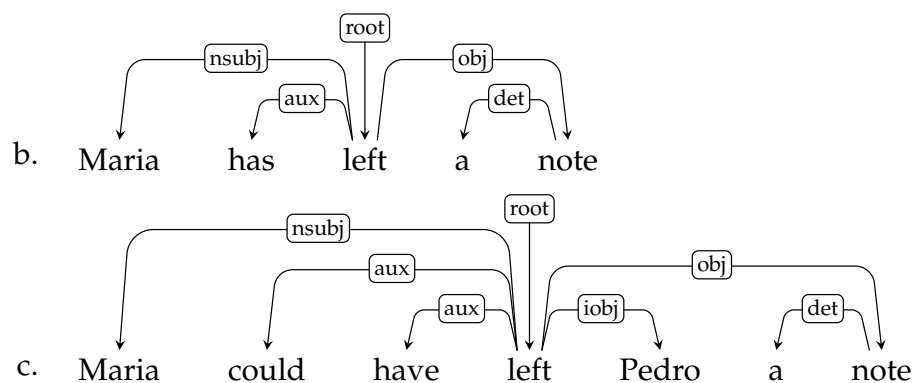
There are a number of additional structures that may appear in nominals, including compounding and flat structures (see Section 3.4.1 and 3.4.3, respectively), and clausal modifiers, particularly in relative clauses (see Section 3.3.2).

3.2 Clauses

A clause consists of a predicate together with its core arguments and oblique modifiers. In this section, we focus on *simple* clauses where dependents of the predicate are nominals, adverbs, or function words. Complex clauses, where a subordinate clause acts as a core or oblique dependent, are discussed in Section 3.3.

3.2.1 Predicates and Core Arguments. In most clauses, the main predicate is a verb, which can be intransitive, transitive, or (in some languages) ditransitive, as illustrated in Example (16a–16c).

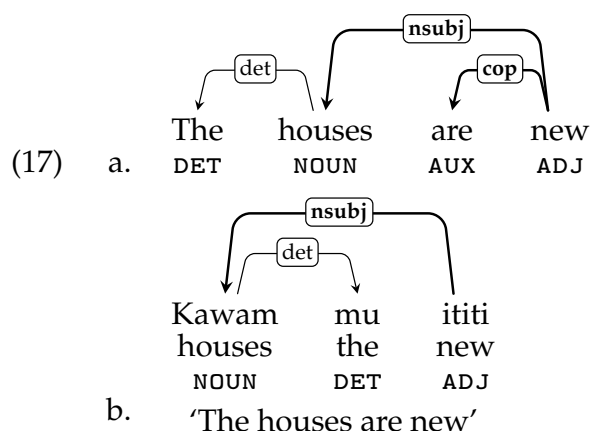




In Example (16a), the intransitive verb *left* has a single core argument, a nominal subject (*nsbj*). In Example (16b), the verb takes an additional core argument, a direct object (*obj*). In Example (16c), finally, there is a third core argument analyzed as an indirect object (*iobj*). In English, nominal core arguments are never introduced by prepositions. Therefore, in a sentence like *Maria could have left a note for Pedro*, the prepositional phrase *for Pedro* is analyzed as an oblique nominal dependent (*ob1*) despite its near semantic equivalence to *Pedro* in Example (16c). We introduced the problem of identifying core arguments in Section 2.3.2 and will return to its crosslinguistic application in Section 4, after we have completed the overview of grammatical constructions in UD.

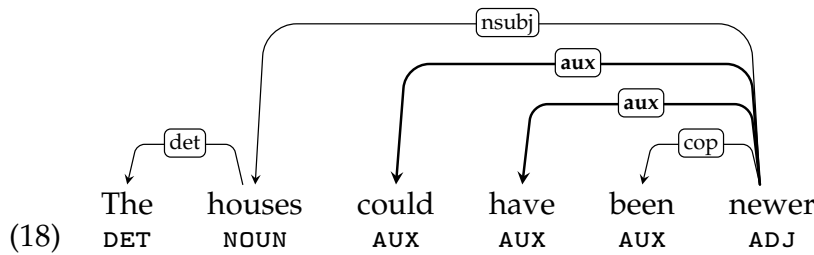
Examples (16b) and (16c) also illustrate that auxiliary verbs are treated as dependents of main verbs in UD with the *aux* relation. Auxiliary verbs help specify verbal features such as tense, aspect, modality, evidentiality, or voice. They may also carry agreement features that cross-reference the subject or other core arguments. The criteria for distinguishing auxiliaries are again language-specific, but auxiliaries are always a closed class and usually a small one. If there are multiple auxiliaries in one clause, a flat structure is created where all auxiliaries are attached directly to the main verb.

Another basic clause construction is that of nonverbal predication, where the main predicate is not a verb but a noun or adjective which usually takes a single core argument analyzed as a nominal subject (*nsbj*). This is a common construction in most languages, but languages differ in the strategies they use to realize the construction morphosyntactically. This is illustrated in the examples below, which show equivalent sentences in English Example (17a) and Waskia Example (17b).



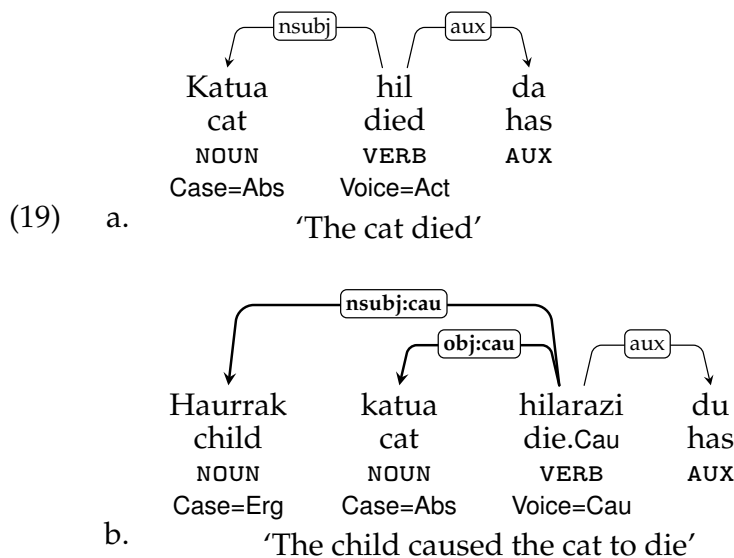
English uses a copula strategy, with a special verb linking the predicate to its subject, while Waskia uses a zero strategy, with no overt linker. By attaching the subject to the nonverbal predicate in both cases, UD highlights the similarity of the construction across languages with different realization strategies. The copula verb is attached to

the nonverbal predicate with the *cop* relation. Other auxiliaries are attached to the nonverbal predicate with the *aux* relation, as in Example (18).



Many languages have ways of altering the mapping between the grammatical relations and the semantic roles of a verb. Such transformations involve changing the form of the verb (using morphology, auxiliaries, or both) as well as the encoding of its dependents. For example, in the **passive** construction the original object is promoted to subject, while the original subject either disappears or is demoted to an oblique modifier. The UD analysis acknowledges the new grammatical relations of dependents, and in this case labels them as subject and oblique, respectively. Nevertheless, to signal that the mapping from grammatical relations to semantic roles has changed, UD provides the subtype *nsubj:pass* for the passive subject (and the subtype *obl:agent* for the oblique modifier). In addition, a passive auxiliary will be labeled *aux:pass* and a morphologically inflected verb will carry the feature *Voice=Pass*. Examples of passive constructions can be found in Example (8) and Example (9).

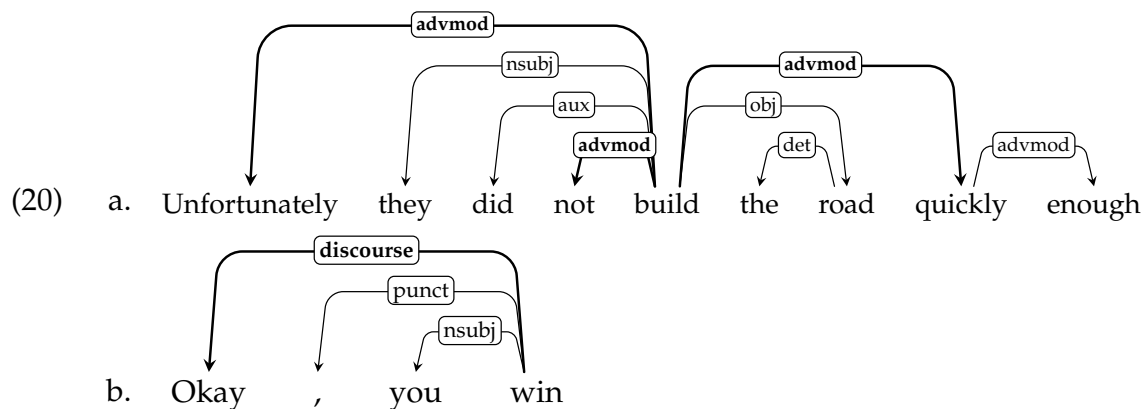
While the passive removes a core argument, the **causative** construction instead adds a new core argument. In the Basque examples below (Oyharcabal 2003), the intransitive sentence Example (19a) is converted to the transitive Example (19b). Here the subtypes *nsubj:cau* and *obj:cau* are used to signal the extended valency frame.



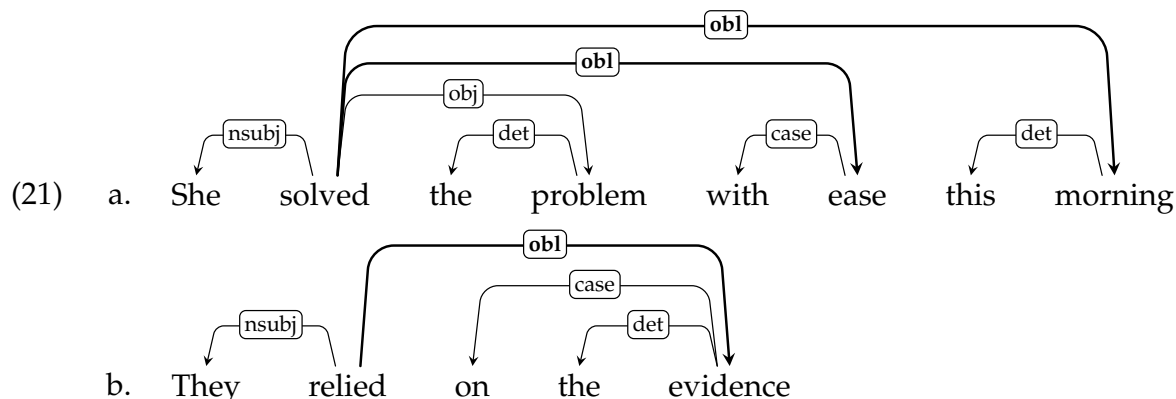
Other valency-changing operations are antipassive, applicative, and the symmetric voice in Western Austronesian languages. For broader typological considerations of voice, see Section 4.

3.2.2 Oblique Modifiers. While predicates and their core arguments form the backbone of a clause, predicates can also be modified in a number of different ways. A large and relatively heterogeneous class of modifiers consists of adverbs, which modify either the predicate or the entire clause with respect to categories such as manner (*quickly*

in Example (20a)), polarity (*not* in Example (20a)), and speaker attitude (*unfortunately* in Example (20a)). All of these modifiers are attached to the main predicate with the `advmod` relation. For discourse particles and interjections, the `discourse` relation is used, as illustrated in Example (20b).



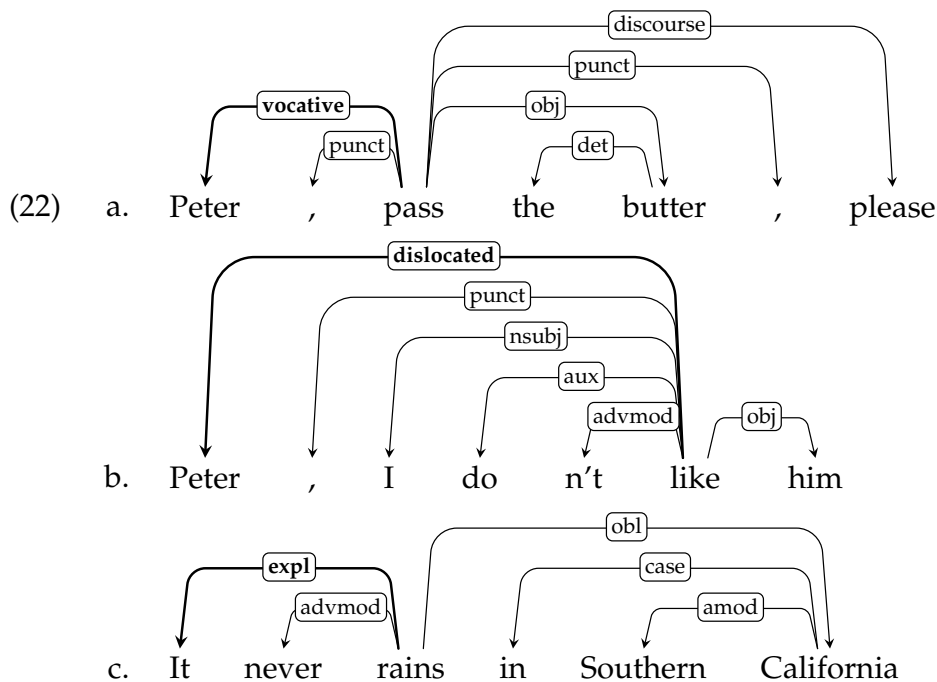
In addition to adverbs and discourse particles, oblique modifiers may also appear in the form of nominals. The `obl` relation is reserved for nominals that are dependents of clausal predicates¹¹ but do not satisfy the criteria for being core arguments. This includes not only nominals whose function is similar to adverbial modifiers, like *with ease* and *this morning* in Example (21a), but also nominals that are arguments semantically, like *on the evidence* in Example (21b). The criteria for distinguishing the latter type from core arguments is discussed in more detail in Section 4.



The `obl` relation covers most non-core nominal dependents of predicates, but there are three special cases for which other relations are used, exemplified below. First, vocatives are nominals that are directed to an (imagined or real) addressee, as in Example (22a). They are attached to the main predicate with the `vocative` relation. Note that the vocative is not the subject of the imperative clause, even if it happens to refer to the actor of the event (and a vocative could equally well occur in a declarative sentence or a question). Second, dislocated nominals are nominals that occur peripherally (initially or finally) in a clause and that serve to contextualize or emphasize a participant of the clause. They do not fulfill a core argument role in the clause but often have discourse

¹¹ To be precise, oblique nominals and adverbial modifiers are used for modification of non-nominals, including modification of adjectives and adverbs that are not clausal predicates. Because adjectives and adverbs normally act as modifiers and their own modification is possible but infrequent (Section 2.1.2), UD reuses the modifier phrase type and the relations `advmod` and `obl` rather than defining new relation types.

prominence, such as being a topic, and are usually anaphorically related with a core argument. The relationship is often coreference, such as in Example (22b), where the nominal *Peter* introduces a topical referent, which is then picked up anaphorically by the nominal object *him*, but there are also cases of bridging anaphora, such as the Japanese topic Example (62) in Section 4.3. Third, expletives are pronominal forms that occur in a core argument position but are not assigned any semantic role. A typical example is the dummy subject of a weather verb, which occurs in English and other languages that require the subject position to be filled in (non-imperative) clauses, as exemplified in Example (22c).¹²



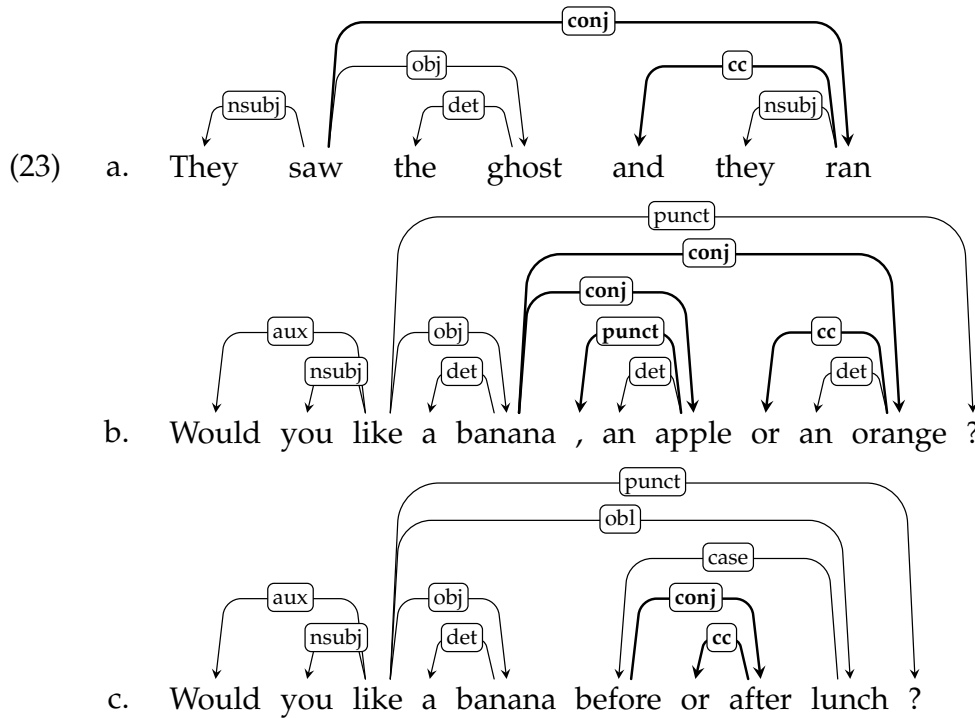
3.3 Complex Constructions

In this section, we describe a variety of linguistic structures, which have in common that they involve clauses embedded into larger structures through relations of coordination or subordination.¹³ It will not be possible to survey this class of constructions exhaustively, so the emphasis is on illustrating the general principles underlying their treatment in UD.

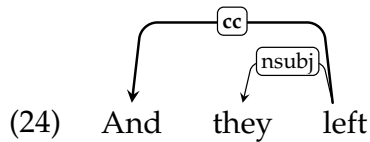
3.3.1 Coordination. All cases of coordination, at the clause Example (23a), phrase Example (23b), or word Example (23c) level, receive the same analysis. UD in principle assumes a symmetric relation between conjuncts, which have equal status as syntactic heads of the coordinate structure. However, because the dependency tree format does not allow this analysis to be encoded directly, the first conjunct in the linear order is by convention always treated as the parent of all other conjuncts. Coordinating conjunctions and punctuation delimiting the conjuncts are attached to an adjacent conjunct using the *cc* and *punct* relations, respectively.

¹² A detailed discussion of different expletives and their treatment in UD can be found in Bouma et al. (2018).

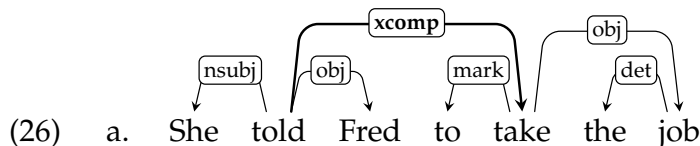
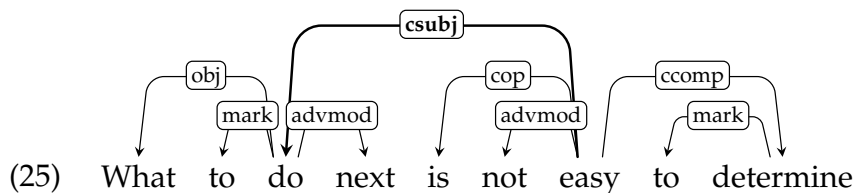
¹³ The only exception is phrase and word-level coordination, which is discussed together with clausal coordination in Section 3.3.1 for convenience.

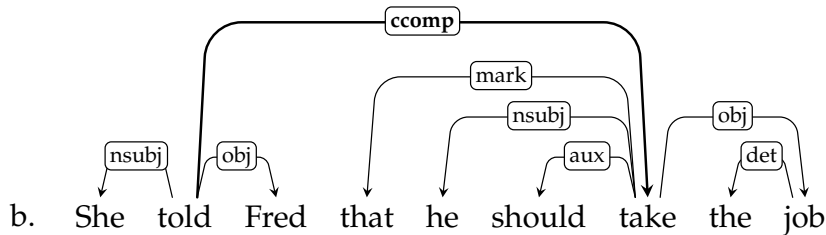


As pointed out by Gerdes and Kahane (2016), the attachment choice of the coordinating element to an adjacent conjunct is motivated by structural properties in many languages, because they together constitute a phrase. Furthermore, such an analysis can provide a parallel analysis for sentences introduced by a conjunct as in Example (24).

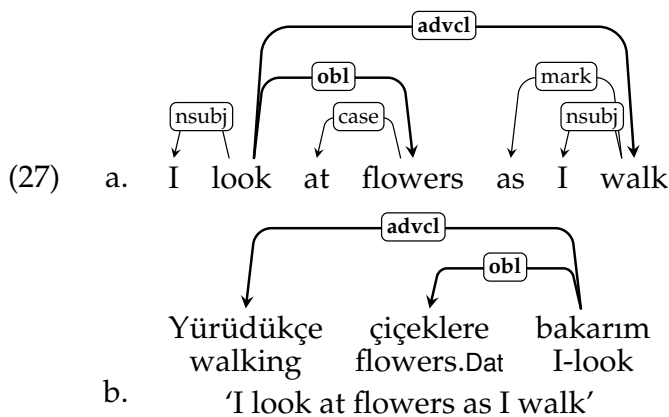


3.3.2 *Subordination*. UD distinguishes four types of subordinate clauses: clausal subjects (csubj) as in Example (25); clausal complements (objects), divided into those with obligatory subject control (xcomp) as in Example (26a) and those without (ccomp) as in Example (26b); adverbial clause modifiers (advcl) as in Example (27); and adnominal clause modifiers (ac1), with relative clauses as an important subtype in many languages Example (28).

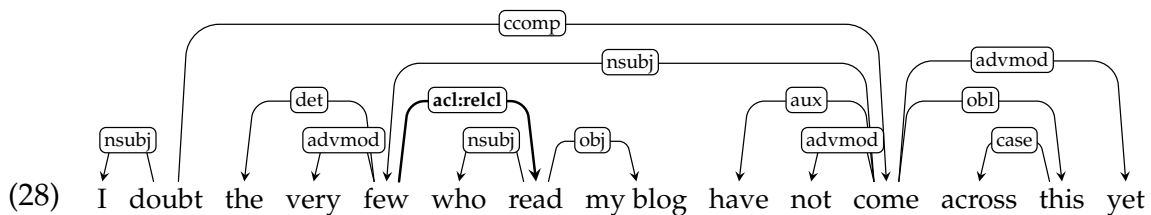




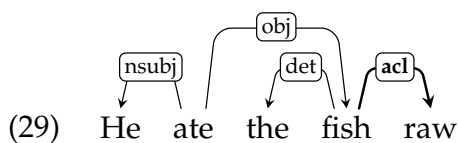
Following the principle of prioritizing relations between content words, the head of a subordinate clause is its predicate, while markers of subordination (e.g., subordinating conjunctions), if any, are attached to the head of the clause they are in, with the relation *mark*. This leads to parallel analyses in English and in Turkish despite different strategies for expressing the subordinated clause: The adverbial clause in English Example (27a) is introduced by the subordinating marker *as* where Turkish uses the morphological marker *-çe* Example (27b).



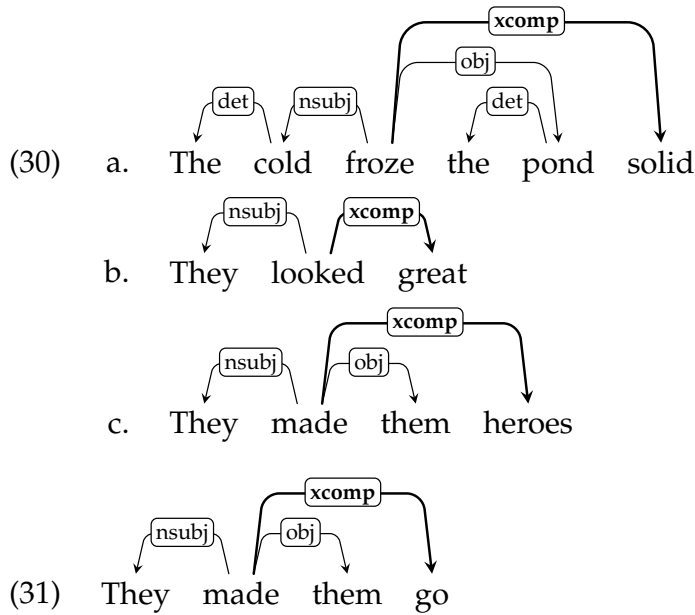
In the case of relative clauses as in Example (28), relative pronouns are attached to the head of the relative clause with the relation corresponding to their grammatical function in that clause (e.g., *nsubj*, *obj*, *obl*).



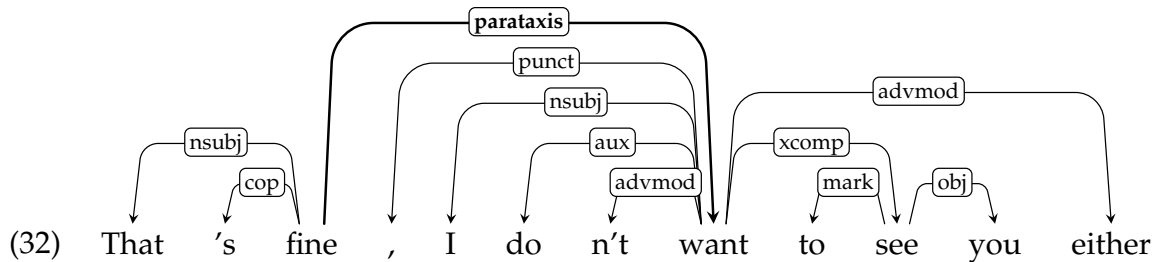
The *acl* relation is also used for optional depictives, such as Example (29), which are thus analyzed as reduced non-verbal clauses, modifying a nominal.



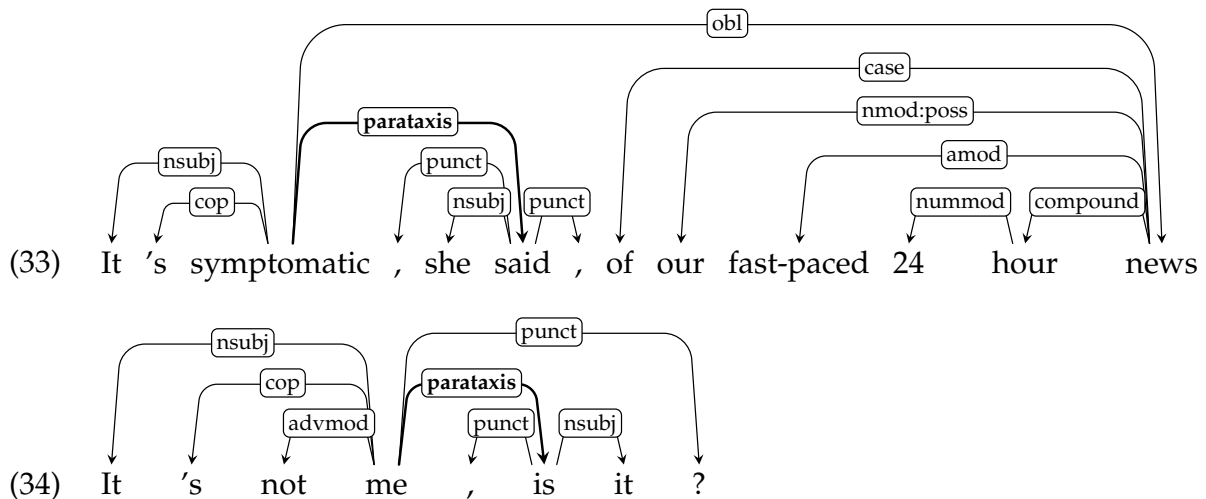
All other secondary predicates (see Huddleston and Pullum [2002] ch. 4), optional resultatives Example (30a), as well as obligatory depictives Example (30b) and obligatory resultatives Example (30c), are treated as core arguments, following Huddleston and Pullum (2002), and given an *xcomp* analysis. UD adopts the same analysis for small clauses, such as Example (31), which share properties of obligatory secondary predicates.

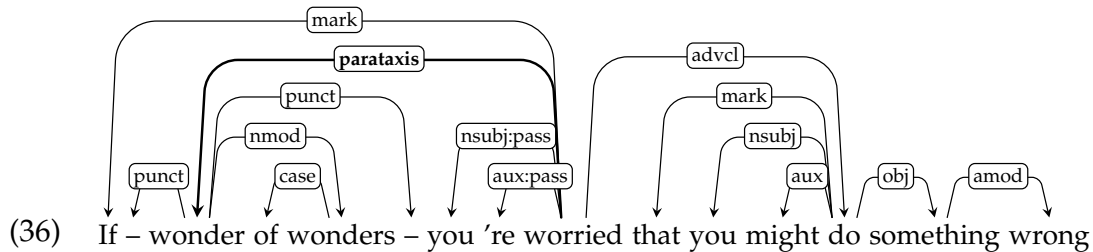
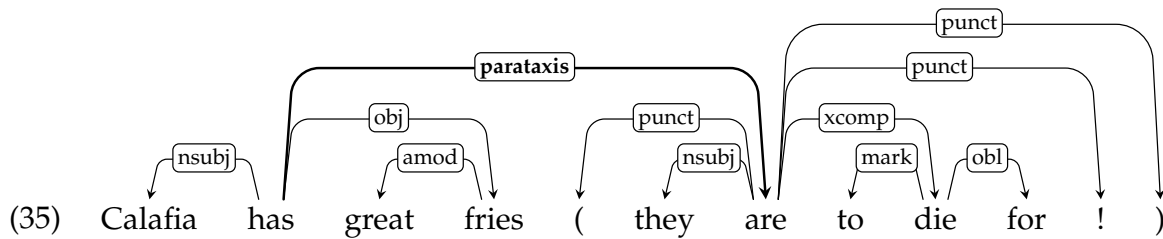


3.3.3 *Parataxis*. UD introduces the parataxis relation to capture clauses or other constituents placed side by side without any explicit coordination or subordination, as in Example (32). This subtype of parataxis can be viewed as a discourse-like equivalent of coordination—whether or not there is punctuation (comma, semi-colon, or colon)—and therefore we follow the same convention as coordination, with the first constituent being the parent.



Some other constructions are also given a parataxis analysis: reported speech Example (33), tag questions Example (34), interjected clauses Example (35), or interjected constituents Example (36). In these cases, the added material is the parataxis dependent (and the parent does not necessarily occur before the child).

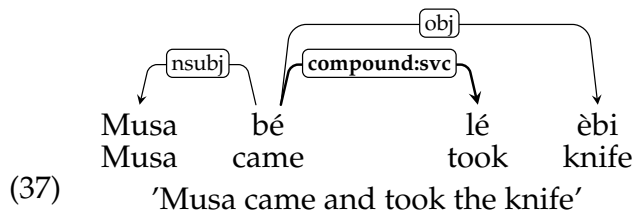




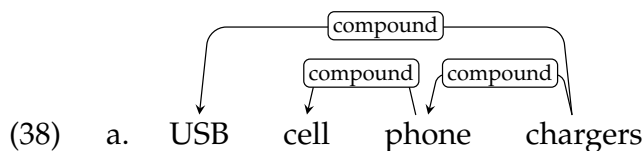
3.4 Multiword Expressions

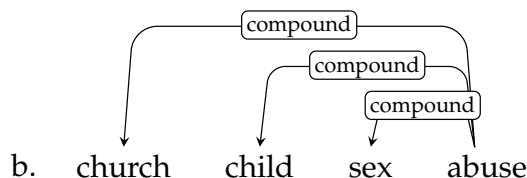
The most regular process of sentence construction in human languages is for a word to be able to take arguments and modifiers that themselves allow further expansion with their own modifiers. For example, *house* can take a modifier like *decrepit*, but that modifier can take its own modifiers and you can form an expression such as [*really rather decrepit*] *house*. However, languages also include constructions where multiple words form a compound or fixed expression. Under a lexicalist approach, such multi-lexeme units are fundamentally different from cases of phrasal modification. UD provides three relations to capture multiword expressions (MWEs), suggesting that these capture the main distinctive groups of MWEs.

3.4.1 *Compound*. The first, and best recognized, situation is compounding. The relation *compound* is used for any kind of word-level compounding: noun compounds (e.g., *phone book*), but also verb and adjective compounds, such as a Japanese light verb construction, such as *benkyō suru* ‘to study’, or the serial verbs that occur in many languages, such as this Nupe example:

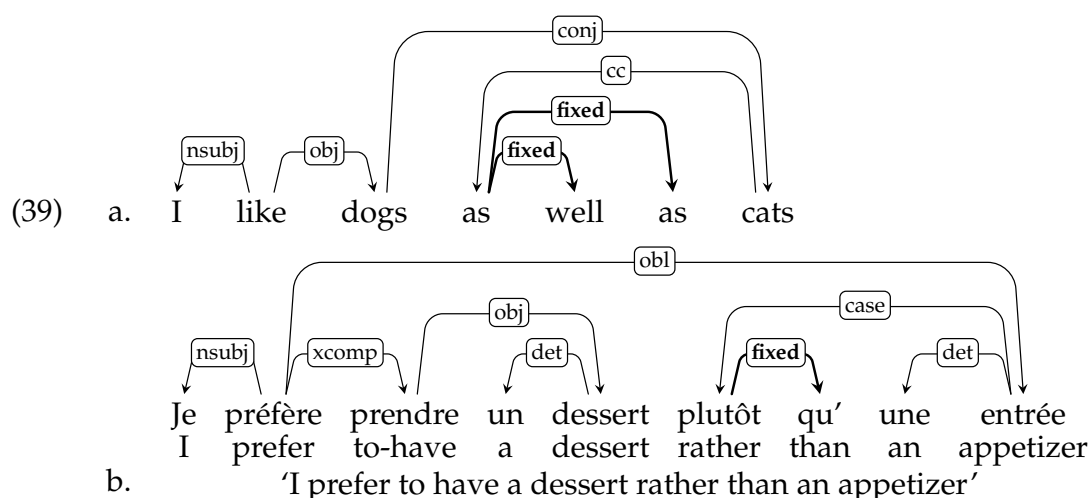


The compound relation is also used for phrasal verbs, such as *put up*: The adverb *up* is attached to *put* via *compound:prt*. Compounds are seen as regular headed constructions: The compound modification relationships indicate the structure of the compound, as shown in Example (38). This behavior distinguishes compounds from the other two types of MWEs.



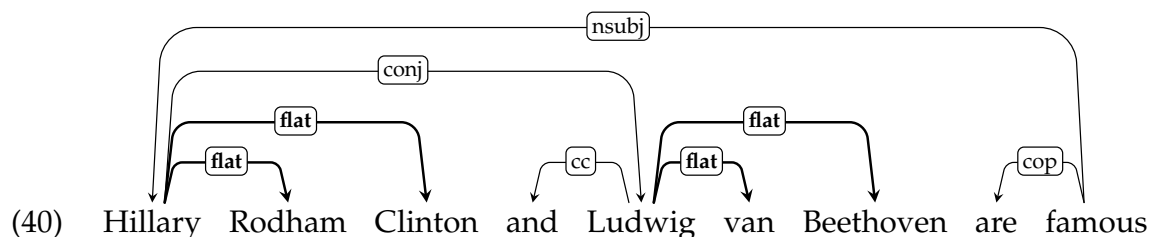


3.4.2 *Fixed*. The *fixed* relation is used for highly grammaticalized expressions that typically behave as function words or short adverbials. The name and rough scope of usage is borrowed from the fixed expressions category of Sag et al. (2002). Fixed MWEs are annotated with a flat structure. Because there is no clear basis for internal syntactic structure, we adopt the convention of always attaching subsequent words to the first one with the *fixed* label Example (39).



As with other clines of grammaticalization, it is not always clear where to draw the line between giving a regular syntactic analysis versus a fixed expression analysis of a conventionalized expression. In practice, the best solution is to be conservative and to prefer a regular syntactic analysis except when an expression is highly opaque and clearly does not have internal syntactic structure (except from a historical perspective).

3.4.3 *Flat Multiword Expressions*. The final class of MWEs is *flat*. This class is less clearly recognized in most grammars of human languages, but in practice there are many linguistic constructions with a sequence of words that do not have any clear synchronic grammatical structure but are not fixed expressions. These include names without internal syntactic structure, and calqued expressions from other languages. We again adopt the convention that in these cases subsequent words are attached to the first word with the *flat* relation, as in Example (40).

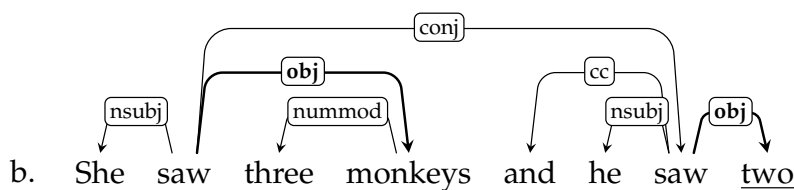
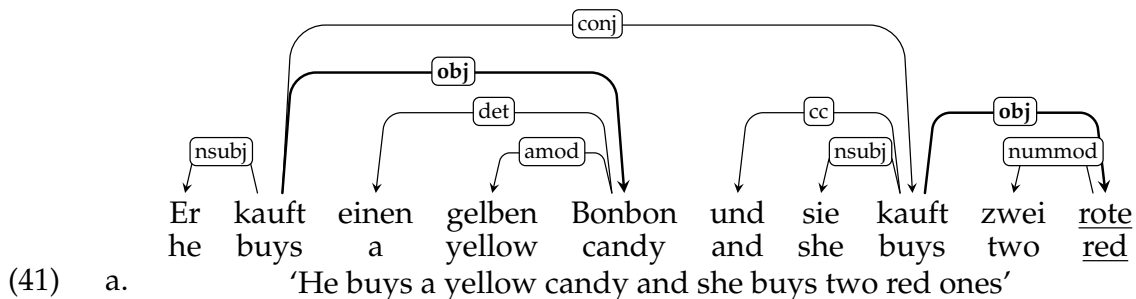


3.5 Ellipsis

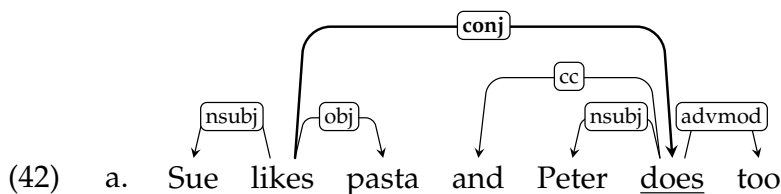
The analysis of ellipsis poses a challenge for all linguistic theories, especially those that do not make use of null nodes (or empty categories) to represent non-overt linguistic elements. UD adopts a compromise solution in this respect. The strategy for analyzing ellipsis is to preserve as many dependency relations as possible and resort to a special relation, which explicitly marks the ellipsis, only when absolutely necessary. The representation discussed here is restricted to overtly realized elements.¹⁴ The strategy is realized as follows:

- If the elided element has no overt dependents, nothing is done.
- If the elided element has overt dependents, one of these is *promoted* to the role of the head.
- If the elided element is a predicate and the promoted element is one of its arguments or phrasal modifiers, the special orphan relation is used when attaching other non-functional dependents to the promoted head.

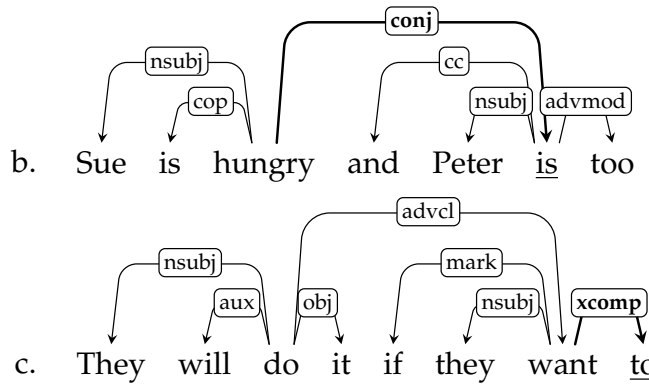
3.5.1 *Ellipsis in Nominals.* If a nominal head is elided, dependents are promoted as head in the following priority order: amod > nummod > det > nmod > case. In German Example (41a), the amod (*rote* ‘red’) of the elided noun (*Bonbon* ‘candy’) is promoted; in Example (41b), the nummod (*two*) is.



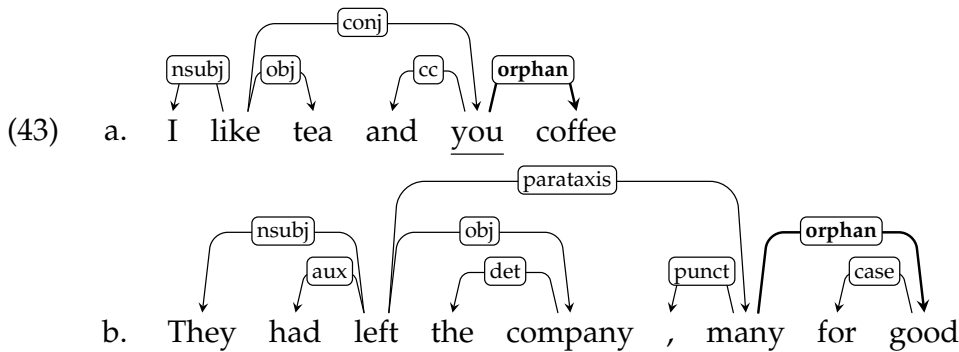
3.5.2 *Ellipsis in Clauses.* If the main predicate of a clause is elided, the aux, cop, or a mark (in the case of an infinitival marker) dependents of the elided predicate are promoted, as illustrated in Example (42a), Example (42b), Example (42c), respectively.



¹⁴ In some cases, null nodes are used in the *enhanced* representation to better capture the predicate–argument structure.



If there is no *aux* or *cop* to promote (or *mark* in the special case of infinitives), dependents are promoted in the following priority order: *nsubj* > *obj* > *iobj* > *obl* > *advmod* > *csubj* > *xcomp* > *ccomp* > *advcl* > *dislocated* > *vocative*. However, to avoid confusion and to signal that the dependency structure is incomplete, the special orphan relation is used to connect the non-promoted dependents to the promoted dependent, as exemplified in Example (43).



Note that the orphan relation is only used when an ordinary relation would be misleading (for example, when attaching an object to a subject). In particular, the ordinary *cc* relation should be used for the coordinating conjunction, which attaches to the pseudo-constituent formed through the orphan dependency, as shown in Example (43a) above, and similarly for the *punct* relation in Example (43b).

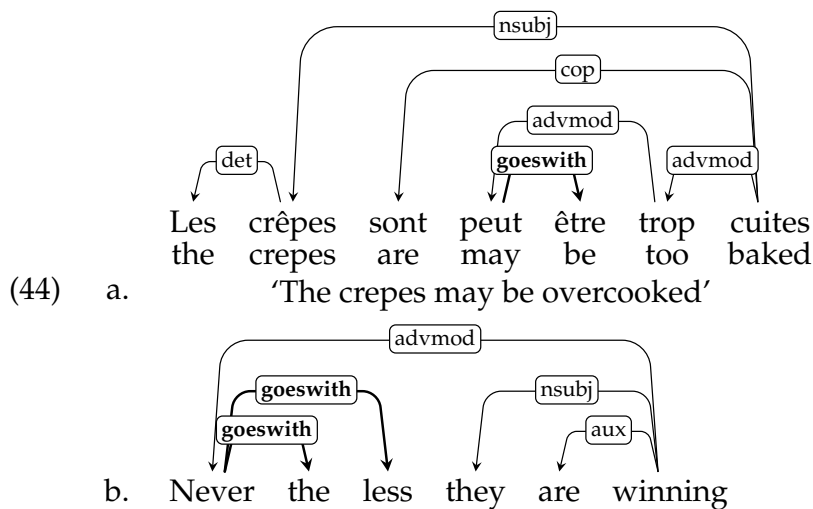
Using the orphan relation in cases of predicate ellipsis results in a severely under-specified predicate–argument representation but prevents the construction of a completely misleading dependency structure, where core argument and modifier relations are used to link words that are really co-dependents.

3.6 Miscellaneous Constructions Found in Corpora but Not Usually in Grammar Books

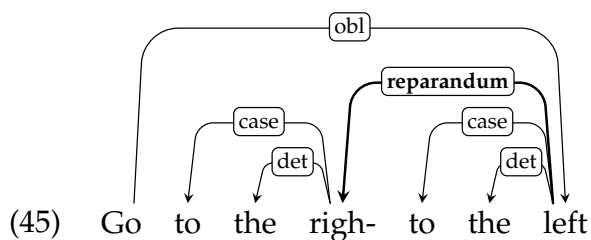
The application of the UD framework to naturally occurring data revealed the existence of several highly frequent constructions that are not discussed in comprehensive grammars. We give examples here, and the analysis proposed under the UD framework.

3.6.1 Special Relations for Informal Genres. Contrary to edited texts, text coming from informal genres, such as Web forums and social media data, and from speech transcripts often contain words wrongly broken into multiple tokens. Examples are given in Example (44a) where the French word for *maybe* is spelled over two tokens but should be one (*peut-être*), and in Example (44b) where the English word *nevertheless* is split into three

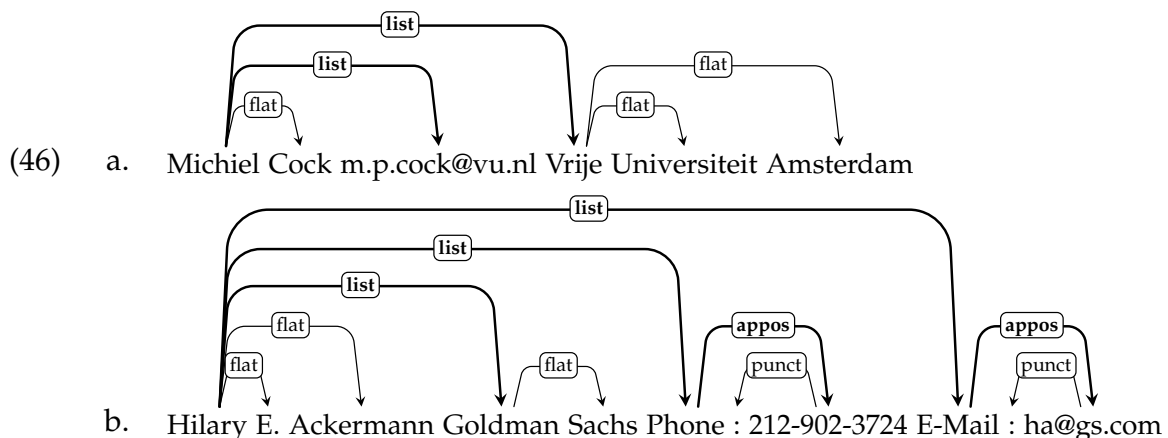
tokens. UD does not assume that a tokenization or normalization process can fix all these errors, and therefore provides a relation, *goeswith*, to indicate that these tokens should be seen as one word. Analogously to the *fixed* and *flat* relations, we adopt the convention of always attaching subsequent tokens to the first one.



Similarly, transcripts contain speech repairs. UD uses the *reparandum* relation to indicate such disfluencies. The repair is chosen as the head because it constitutes the final utterance, with the disfluency being the dependent of the repair, as shown in Example (45).



3.6.2 Lists. When dealing with Web data, we frequently encounter passages, parsed as single sentences, that are meant to be interpreted as lists. Email signatures are a typical example of such lists, as in Example (46a). UD uses the *list* relation to link the different elements, with the first one being the head. In some cases, the fields in the list are explicit, and take the form of a “key:value” structure. UD uses the *appos* relation to link a value to its key, as in Example (46b).

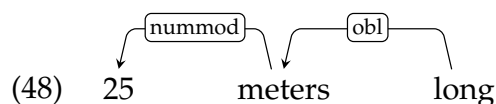


3.6.3 *Noun + Number/Letter Constructions*. Another frequent construction in all UD corpora is a noun followed by a number or a letter (or a combination of both), such as in the English examples in Example (47).

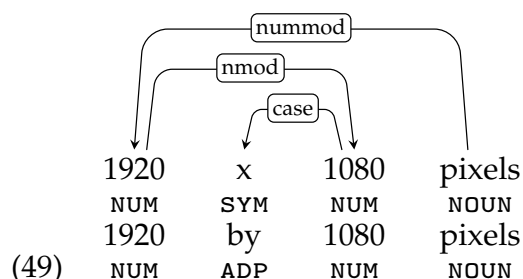
- (47) a. This is the *number one* restaurant in town
 b. He lives on *floor four*
 c. *Bus 102L* takes you straight to the center
 d. On *day 2* of our trip, we hiked to the bottom of the canyon
 e. The meeting will be in *room A*

For a uniform treatment across such constructions, UD treats them as noun–noun constructions. While some of the examples above have an ordinal reading, such as Example (47b) or Example (47d), where the expressions can be paraphrased respectively as *on the fourth floor* and *on the second day*, UD analyzes the number as a noun to maximize the parallelism with constructions that use a letter or a combination of both number and letter; indeed, one can live *on floor C* where *C* acts as a noun. Therefore the number/letter expression attaches to the noun it modifies via a `nmod` relation, unless there is clear morphosyntactic evidence in the language for the opposite direction.

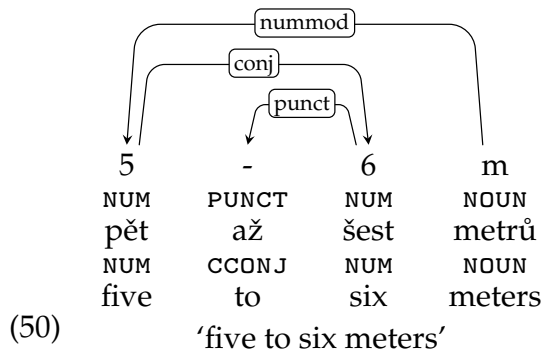
3.6.4 *Measure Phrases*. The analysis of simple measure phrases, such as *5 years old* or *25 meters long*, is relatively straightforward, illustrated in Example (48): The number serves to modify the meaning of the noun with a quantity and the measure noun is seen as functionally corresponding to an adverbial modifying the adjective.



There are also complex measure phrases involving symbols, such as *1920 × 1080 pixels*, or ranges (*5 – 6 meters*). In such cases, the UD analysis follows the reading of the expression in the language. For instance, in the English Example (49), the symbol acts like a preposition *by* and is analyzed as such.



In some cases like in the Czech Example (50), the symbol is pronounced as a coordinating conjunction. It is thus analyzed as a punctuation `PUNCT` (see Section 3.3.1 on coordination) and the numerical constituent as a coordination.



4. Core Grammatical Relations: A Typological Perspective

One of the main challenges for a framework like UD is to ensure that universal categories are applied consistently across languages with sometimes radically different morphosyntactic encoding strategies. This can only be achieved through a complex interplay between abstract language-independent guidelines and concrete language-specific criteria. In this section, we will outline how this idea can be realized for core grammatical relations like subject and object, which play a central role in the UD theory. After stating general criteria derived from the typological literature, we will go through four groups of languages that illustrate different ways of instantiating the general criteria relative to language-specific evidence. The first group is what has been called Standard Average European (Whorf 1956; Haspelmath 2001), which is a homogeneous group but with some subtle differences, exemplified here by English, Czech, and Spanish. The second group is a selection of large non-Indo-European languages—Japanese, Arabic, and Swahili—which introduces more variety in the encoding of core grammatical relations. The third group comprises languages exhibiting different forms of ergativity, a phenomenon that is challenging for any theory based on the notions of subject and object. The fourth group includes languages with voice systems that are substantially different from the active–(middle)–passive that is found in the Indo-European family.

4.1 General Criteria

The starting point is that core arguments can be recognized relatively easily based on surface criteria such as word order, agreement, and case marking (both morphological and syntactic). However, for any given language, one has to first establish which of these criteria apply. For example, many languages have a morphological case called “dative,” but dative nominals act as core arguments in some languages (or uses), and as oblique in other languages (or uses).

To determine which core arguments are available in a given language, and how they are morphosyntactically encoded, it is useful to start with so-called primary transitive clauses (Andrews 2007), that is, clauses with predicates that license the semantic roles of agent (actor) and patient (undergoer) in the prototypical sense. Clauses where the predicate is a verb describing a violent action are often good examples, such as *George killed the dragon*. In such a clause, the predicate has two core arguments: The more active argument (the agent) is said to have the grammatical function A; the other argument (the patient) is said to have the grammatical function P. By observing the coding strategies and grammatical rules that, within the language, are typical for arguments with the functions A and P, we can identify these functions also with other predicates,

regardless of their semantic roles. Such predicates will be called transitive and their arguments will also have the functions A and P, respectively. For instance, *John loves Mary* is a transitive clause, and *John* and *Mary* have the functions A and P, respectively, because the grammar treats them the same way as *George* and *the dragon* in the earlier example. The exact semantic roles are no longer important: John is an experiencer rather than actor, and Mary may not be affected by his love; she may not even be aware of it.

When we can recognize a predicate with two core arguments, we can also recognize predicates that have at most one (regardless of whether they also have additional non-core dependents). Clauses headed by such predicates are intransitive and their single core argument is said to have the grammatical function S. In general, nominals with functions S and A are subjects and labeled *nsubj*, while arguments with function P are objects and labeled *obj*. Finally, some verbs in some languages take three or more core arguments, more than one showing behavior that is characteristic of objects (Haspelmath 2015). Prototypically, such ditransitive constructions involve verbs of giving and transfer, and UD analyzes the theme (i.e., the entity that is transferred) as the direct object, and introduces the relation of indirect object (*iobj*) for the recipient. However, as noted earlier, the *iobj* relation should only be used if the nominal denoting the recipient is encoded as a core argument. In English, for example, this means that the nearly synonymous sentences *Mary gave John a book* and *Mary gave a book to John* differ in that the recipient is realized as an indirect object (*John*) in the former but as an oblique modifier (*to John*) in the latter.

We now discuss how these general principles can be applied to languages with different encoding strategies, starting with familiar Indo-European languages and gradually introducing more diversity.

4.2 Standard Average European

In Indo-European languages with case marking, nominative and accusative cases will usually map to subject and object core arguments, respectively. When there is no case marking, tests based on word order, pronominalization, and passivization can be used to identify core arguments.

English. In English, nominal core arguments are bare nominals (that is, without prepositions) and can be identified, to some extent, using word order. In an unmarked declarative sentence, the core argument preceding the verb is the subject. If there is another core argument following a transitive verb, it is the object, as in Example (16b). English has a remnant of morphological case for some of the personal pronouns: Subject pronouns are in the nominative form (*I, he, she, we, they*) whereas objects are in the accusative form (*me, him, her, us, them*).

The main complication when drawing the core–oblique distinction in English is that, while the presence of a preposition is a sufficient condition for obliqueness, it is not a necessary one. There are bare nominals that are used as oblique (temporal) modifiers, as in Example (51b).

- (51) a. A baker works the dough
 b. A baker works the whole week
 c. John spends the whole week in Paris

The reasons why *the whole week* is not a core argument in Example (51b) (whereas *the dough* and *the whole week* are core arguments in Example (51a) and in Example (51c),

respectively) are complex, but we can use tests based on word order, pronominalization, and passivization to establish that *the whole week* does not behave as a core argument in Example (51b). An oblique modifier (*the whole week* in Example (52a) and Example (52b)) can swap positions with a locational modifier (e.g., *in Paris*), whereas this is not possible for a core argument Example (52c) vs. Example (52d):

- (52) a. John works the whole week in Paris
 b. John works in Paris the whole week
 c. John spends the whole week in Paris
 d. *John spends in Paris the whole week

Second, unlike a direct object, the temporal modifier cannot be pronominalized:

- (53) a. *John worked it in Paris
 b. John spent it in Paris

It is also not possible to promote a temporal modifier to subject by passivization: (**The whole week was worked by John*). This test is not decisive by itself in English, as there are transitive verbs that cannot passivize, and prepositional verbs that can. But taken all together, the tests indicate that *the whole week* in Example (51b) is not an object, and it will therefore attach to the verb with the ob1 relation.

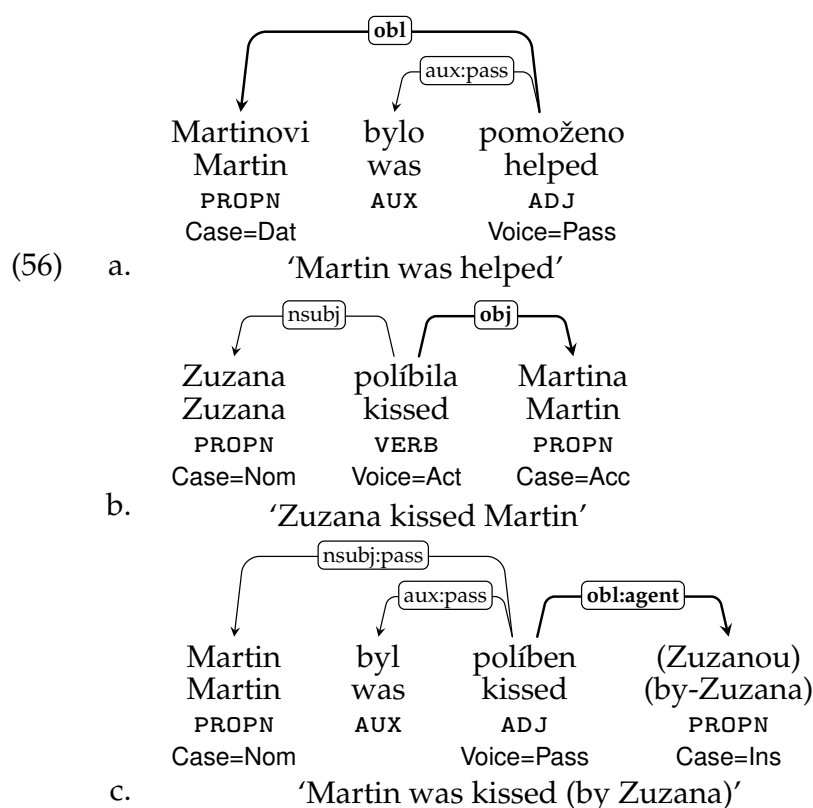
Czech. Czech has substantive morphology that can be used to classify verbal arguments. Core arguments in Czech are bare noun phrases in the nominative for the subject and in the accusative for the object. Whereas SVO order is preferred by default, Czech word order is free: Other permutations are possible and may be required to distinguish topic and focus. Like in English, a bare accusative nominal is not necessarily a core argument. It can be an oblique (temporal) modifier, as *celý týden* ‘whole week’ in Example (54a) or *každou středu* ‘every Wednesday’ in Example (54b).

- (54) a. Pracuje celý týden
 works whole week
 ‘He/she works the whole week’
 b. Přichází každou středu
 comes every Wednesday
 ‘He/she comes every Wednesday’

Many verbs in Czech take, in addition to a subject, a bare noun phrase in a case other than accusative (i.e., in the dative, genitive, or instrumental). UD invariably treats these as oblique (ob1), as in Example (55).

- (55)
- | | | |
|----------|-----------|-----------|
| nsubj | ob1 | |
| └─┬─┘ | └─┬─┘ | |
| Zuzana | pomohla | Martinovi |
| Zuzana | helped | Martin |
| PROPN | VERB | PROPN |
| Case=Nom | Voice=Act | Case=Dat |
- ‘Zuzana helped Martin’

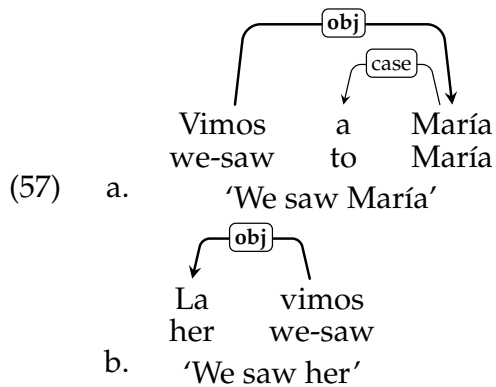
Whether these non-accusative second dependents should be seen as core arguments or not is debatable.¹⁵ There are examples of verbs that take non-accusative second dependents and could be claimed to belong to transitive verbs (viz., *pomohla* ‘helped’ in Example (55)). However, such examples are rare, and non-nominative, non-accusative dependents tend to have semantic roles other than the proto-patient. Also, the treatment of these dependents by grammatical rules such as passivization is different from the treatment that accusatives receive. In Example (56a), which is the passive corresponding to Example (55), *Martinovi* is not promoted to subject: It stays in the dative case, and the passive predicate, instead of cross-referencing Martin’s masculine gender, stays in the default neuter singular form. In contrast, the active sentence Example (56b) features an accusative argument *Martina*, and when passivized in Example (56c), this argument becomes the subject, taking the nominative form and triggering agreement both on the passive participle and on the auxiliary. Thus, UD treats only nominative and accusative dependents as core arguments in Czech.¹⁶



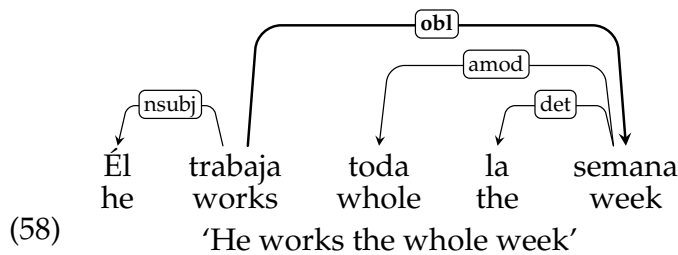
Spanish. Spanish is in many ways similar to English and Czech but does not adhere to the rule that the presence of a preposition is a sufficient condition for obliqueness. Spanish uses the preposition *a* with animate direct objects, as in Example (57a). Such objects, when pronominalized, use the accusative pronoun form Example (57b), and they can be promoted to subjects in passive constructions. Inanimate direct objects behave the same way except that they do not use the preposition. UD therefore treats a nominal with the preposition *a* as a core argument when it is an animate direct object.

¹⁵ For German, Andrews (2007, pp. 182–183) leaves the question open while Foley (2007, p. 377) has no doubt that the dative case is oblique.

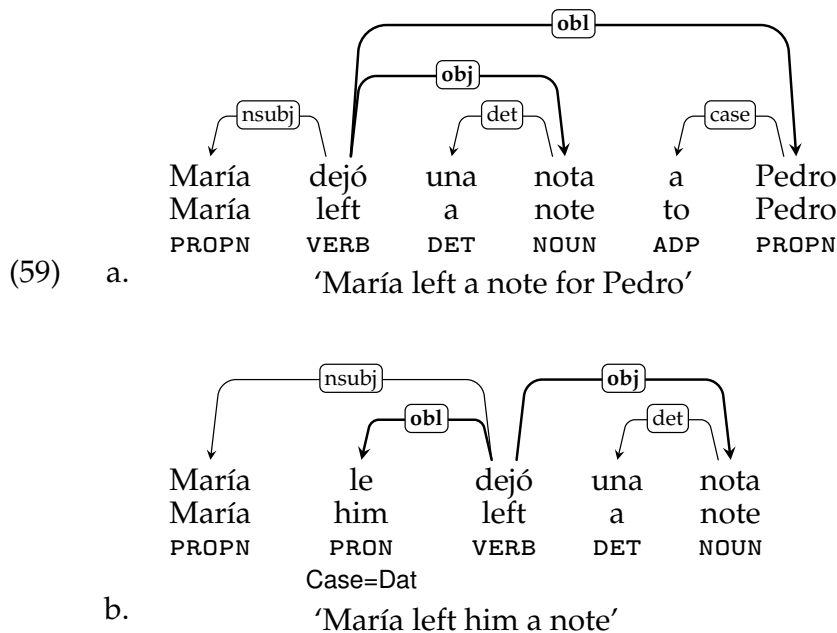
¹⁶ We ignore here certain anomalies in the Czech case system that involve quantified nominals. In the presence of a quantifier, the quantified noun may take the genitive form although the whole quantified phrase occupies a nominative or accusative position.



Similarly to English or Czech, a bare nominal is not necessarily a core argument, again with oblique temporal modifiers being a prime example, as in Example (58).



As mentioned in Section 3.2.1, some languages have two (or even more) object constructions, including Germanic and Bantu languages. For instance, in ditransitive constructions, the predicate has *obj* and *iobj* dependents (see Example (16c) for the English example *María could have left Pedro a note*). Traditionally, Romance languages have been viewed as lacking multiple object constructions, because there is always at most one bare object nominal, and other nominals are expressed with adpositions (as Example (59a) in Spanish).



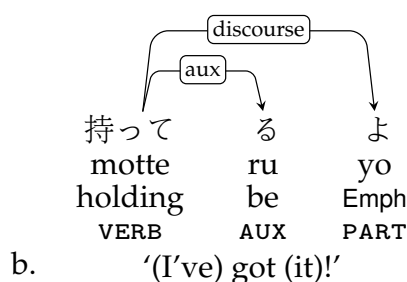
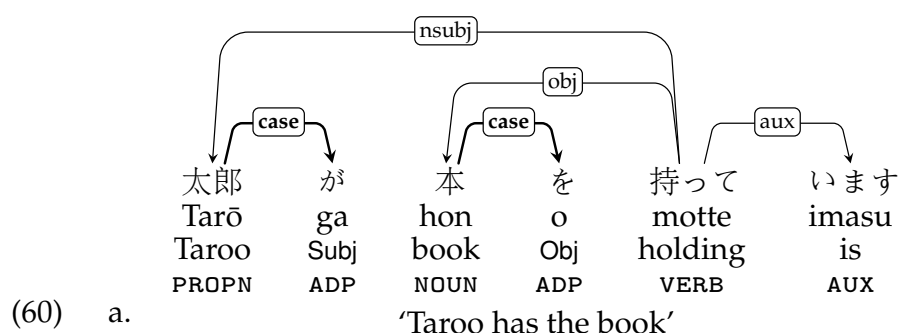
Still the dative seems to have something of a special status. Part of the evidence is the availability of dative clitics, as in Example (59b) (though French also has partitive and locative clitics); other evidence comes from relation-changing operations like causatives. Some people have argued for Romance datives being core arguments (Van Peteghem

2006; Boneh and Nash 2012; Pineda 2013, inter alia) though others have argued against it (Kayne 1984, inter alia).

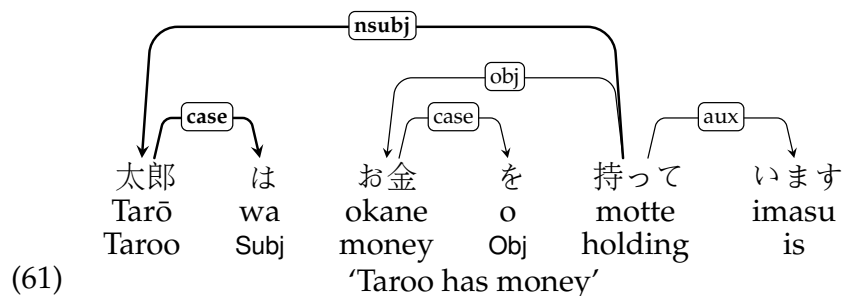
4.3 A Sample of Non-Indo-European Languages

In this section, we extend our discussion of core arguments in UD to three unrelated non-Indo-European languages, each with a large number of speakers: Japanese, Arabic, and Swahili.

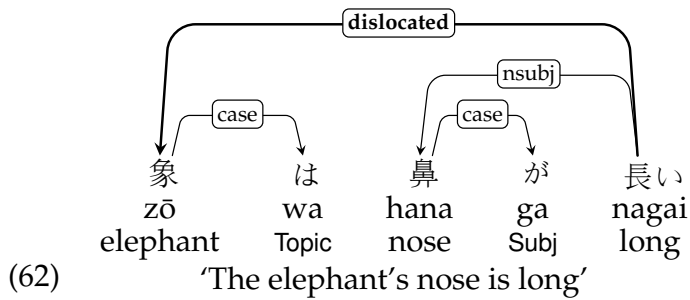
Japanese. In Japanese, while there is a predominant word order, there is considerable word order flexibility and nominal arguments can be freely omitted Example (60b). Grammatical relations are mainly expressed by case particles, which we regard as adpositions bearing the grammatical relation case Example (60a).



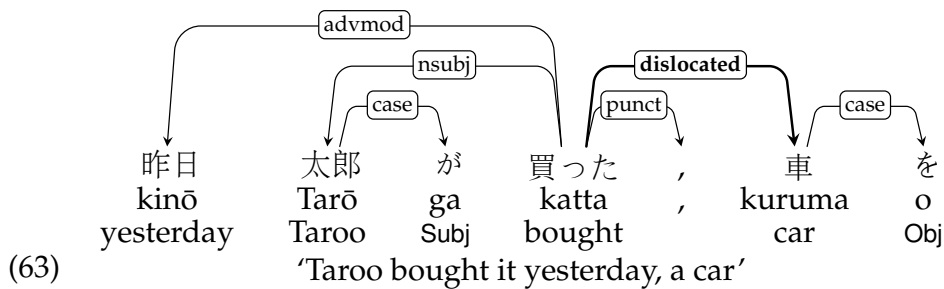
Japanese, like other East Asian languages, is a strongly topic-oriented language. Topics are marked with the case adposition は. Most commonly the topic-marked nominal will be the subject or another regular dependent of the clause, and は will then either replace (for nsubj or obj) or augment (for oblique dependents) the normal case adposition.



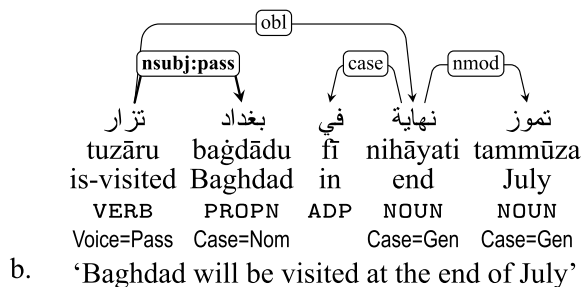
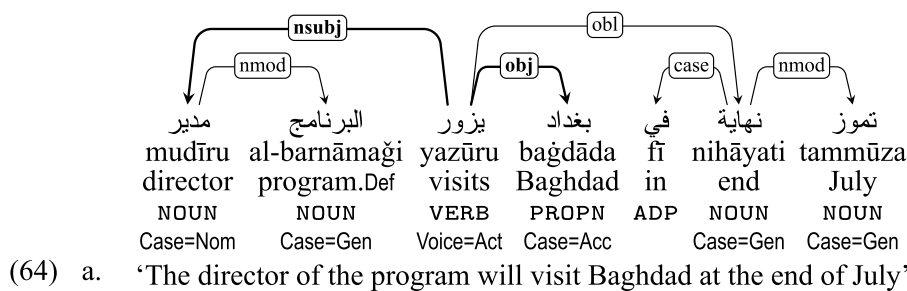
However, a topic may also represent the context of the remainder of the sentence while not being part of the predicate-argument structure. A nominal that establishes a discourse context in this way takes the relation dislocated:



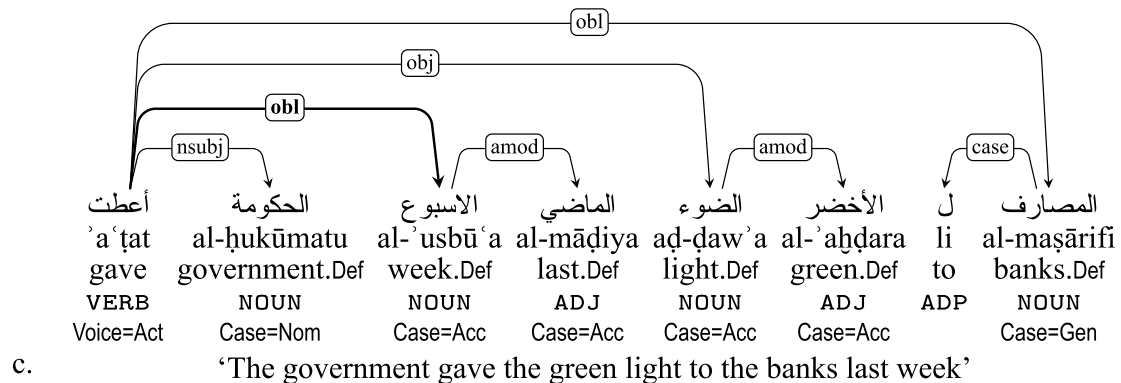
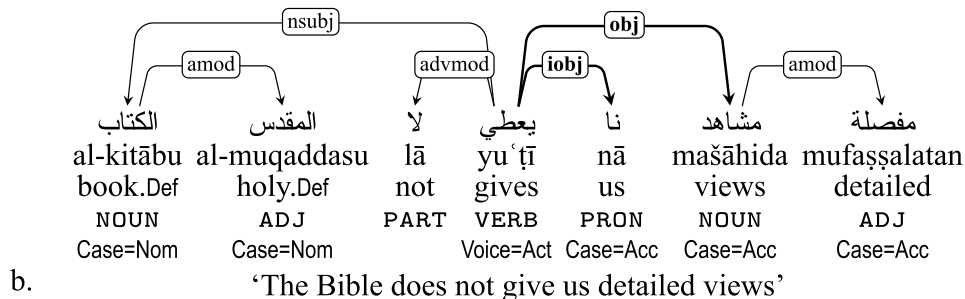
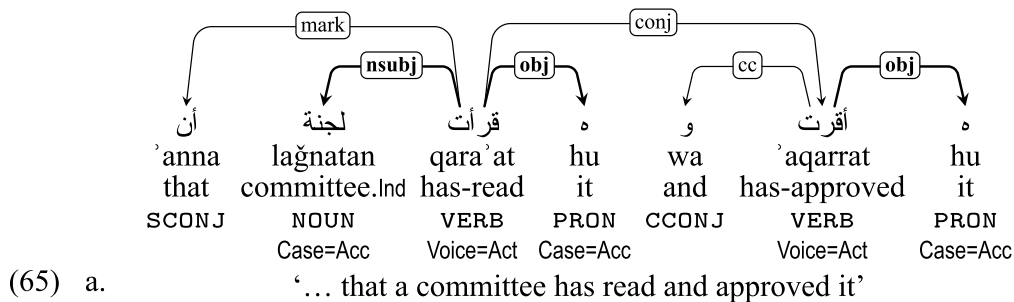
Although basically a head-final language, in spoken Japanese, nominal dependents and nominal dependents of dependents can also sometimes appear after the verb, as a kind of afterthought. These are also treated as dislocated elements:



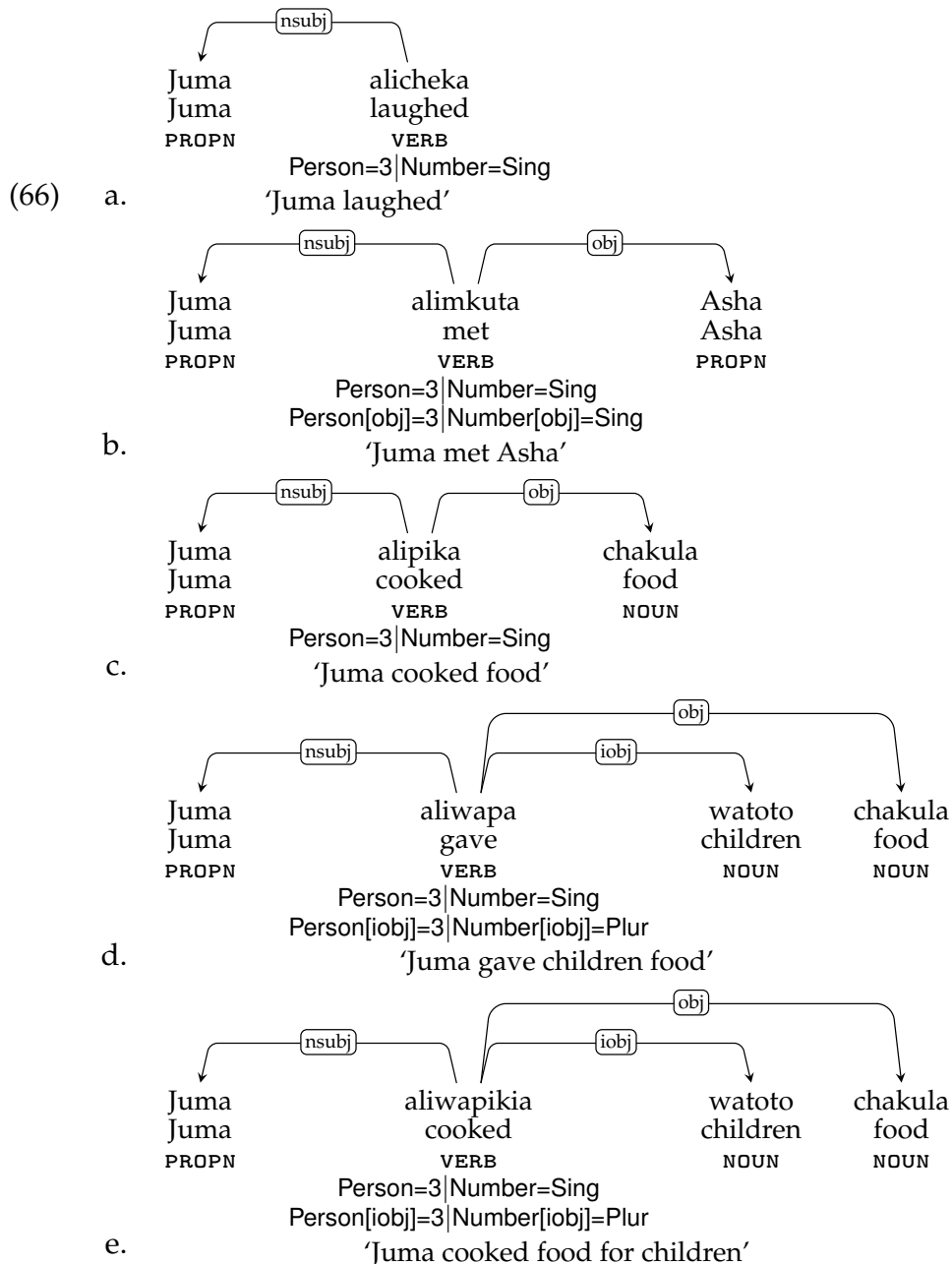
Arabic. Arabic verbs cross-reference the person, number, and gender of their subjects. Nominals are case-marked: The subject is in the nominative, the object in the accusative (except in subordinate clauses with conjunction 'anna 'that' Example (65a), where the subject is also in the accusative). Multiple word orders are possible, subject–verb–object and verb–subject–object being the most frequent. Passive clauses are agentless in Classical Arabic (Fischer 1997, p. 210) but oblique agent phrases are re-introduced in Modern Standard Arabic (Badawi, Carter, and Gully 2013, p. 385). The vowel pattern *a-ū* of the active verb in Example (64a) is replaced by the passive pattern *u-ā* in Example (64b). Furthermore, the masculine prefix *y-* is replaced by feminine *t-* to reflect the gender of the passive subject *baġdādu*.



Subject pronouns can be dropped. Object pronouns are encliticized to the verb Example (65a) but treated as syntactic words in UD. In ditransitive clauses Example (65b), the verb governs two accusative objects; the recipient precedes the theme and, if pronominal, it is encliticized to the verb. Bare accusative nominals are not always core arguments; they can be adjuncts—for example *al-ʿusbūʿa al-mādiya* ‘last week’ in Example (65c). Such ‘adverbial accusatives’ can denote time, location, direction, motivation, manner, and so forth (Fischer 1997, p. 216).



Swahili. In Swahili, core arguments are primarily marked by cross-referencing on the verb. There is no case marking and word order is relatively free, although subjects tend to precede and objects tend to follow the verb. Cross-referencing of the subject is obligatory, as illustrated in Example (66a–66e), where the prefix *a-* consistently marks the subject as third person singular. In transitive clauses, cross-referencing of the direct object is obligatory if it is animate, as in Example (66b) where the prefix *m-* marks the object as third person singular, but optional if it is inanimate, as in Example (66c). In ditransitive clauses, it is the object highest in animacy that is cross-referenced regardless of grammatical relation. Ditransitive clauses may be formed by an inherently ditransitive verb, as in Example (66d), or by an applicative transformation on a transitive verb, as in Example (66e), where the applicative suffix *-i* extends the valency frame of the verb *pik* ‘cook’ with an additional (indirect) object. The fact that the additional dependent is cross-referenced on the verb like any animate object supports its status as a third core argument.

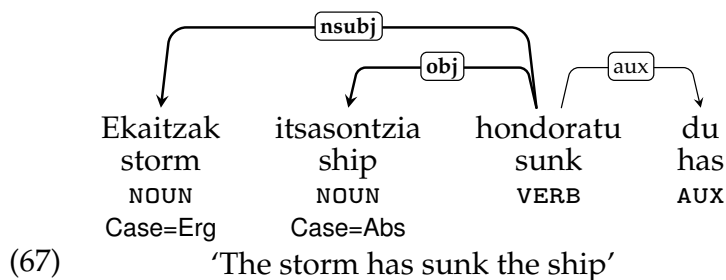


4.4 Ergativity

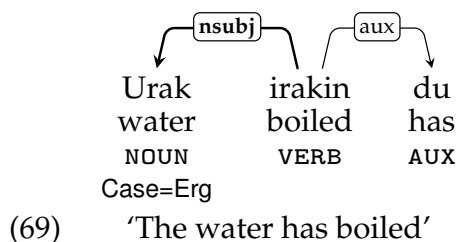
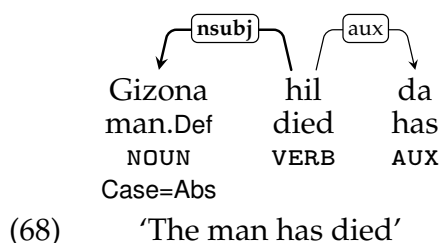
As discussed in Section 4.1, UD generally assumes that the *nsubj* relation covers the grammatical functions S and A, while *obj* is reserved for the grammatical function P. This fits well with the nominative–accusative alignment found in many languages, but it is challenged by the ergative–absolutive alignment that groups S and P together. For many languages, ergative–absolutive case marking appears to be only a morphological feature, which we handle at the level of the Case feature. Basque, below, is an example. For other languages, ergativity has been argued to extend to the treatment of grammatical relations (Dixon 1994). There are then multiple possible analyses (and different ones may apply to different languages). One choice is to regard the ergative as an oblique (Mel'čuk 1988), essentially analyzing all sentences in the language as intransitive, with only one core argument marked in the absolutive, which is used for intransitive arguments and the patient-like argument of transitive verbs. A more frequent analysis is to

say that such syntactically ergative languages treat the intransitive core argument and the patient-like argument of transitives together as a “pivot” (Dixon 1994), which we would analyze as a subject (nsubj), and then the agent-like argument of transitives is also a core argument, which we would analyze as an object (obj). The unusual thing, then, is the reversed alignment between semantic roles and grammatical relations. This is a place where the relation subtype :pass can be usefully used in an extended sense. If we regard it as marking not only passives but all cases where the nsubj does not mark the agent-like argument of the verb, then all transitive subjects in such a language are nsubj:pass. In addition, we can reuse the subtype :agent, which in other languages is optionally used for an oblique modifier denoting a demoted agent, to mark the ergative core argument as obj:agent.

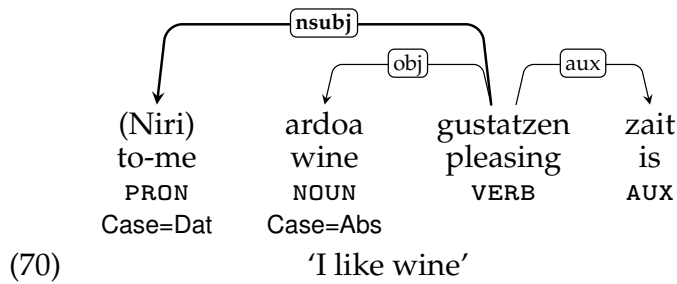
Basque. In Basque (Zúñiga and Fernández 2019), nominal case morphology is the main indicator of core argument relations. However, instead of nominative–accusative, the core pair of cases is ergative–absolutive. Most two-argument verbs have the more agentive argument in the ergative and the patient-like argument in the absolutive case, while single argument verbs usually use the absolutive for their single argument. Nevertheless, there is no evidence that absolutives form a coherent grammatical relation. Rather, the ergative argument is treated as subject (nsubj), while the absolutive argument of transitives is object (obj), as in Example (67).



The single argument of intransitive verbs takes mostly the absolutive Example (68) but sometimes the ergative form Example (69). It is labeled as subject (nsubj) in both cases.

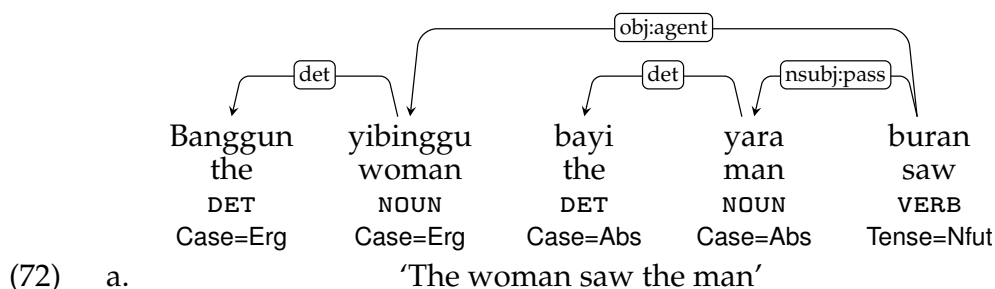
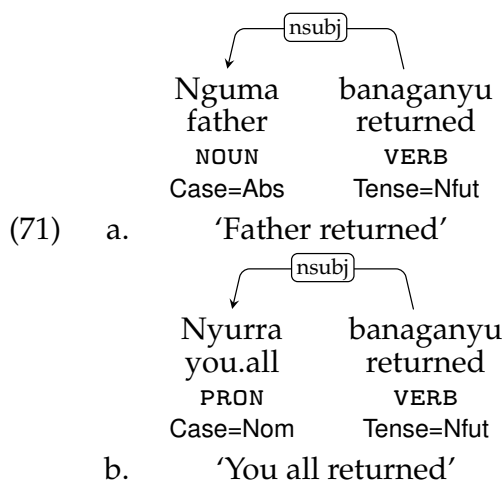


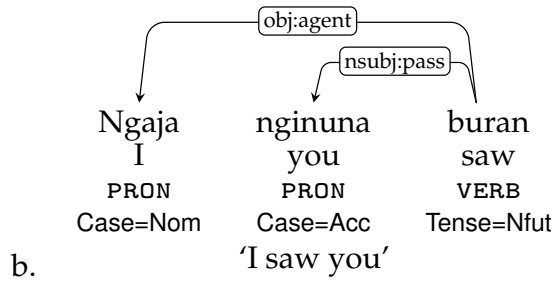
The third core argument case is the dative. Arguments in all three core cases are cross-referenced on finite verbs and can be omitted. Some experiencer-subject two-argument verbs take dative + absolutive, instead of ergative + absolutive, as in Example (70).



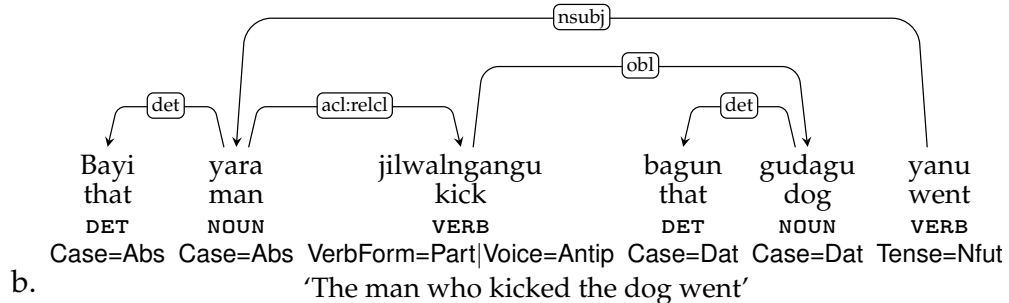
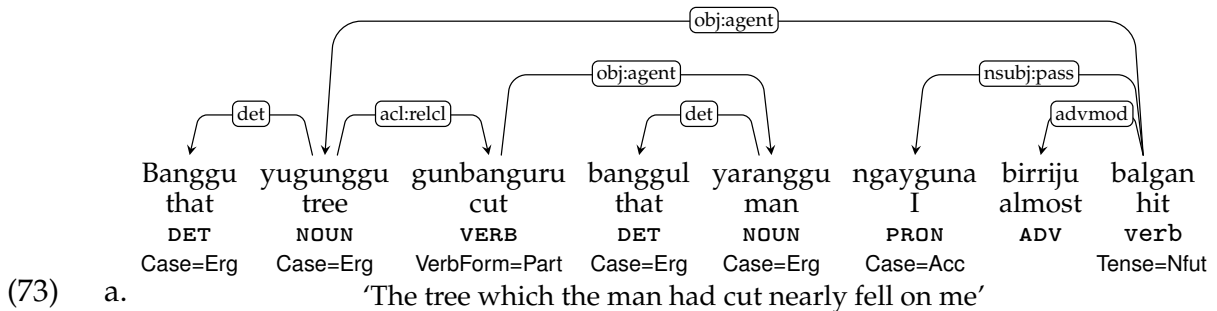
According to Zúñiga and Fernández (2019), the dative encodes the A function in such constructions, which makes it subject in UD. Supporting evidence for this is provided by causativization, a valency-changing operation that takes a transitive clause, adds a third, ergative argument, and switches the original subject to the dative (unless it already was in dative). The fact that causativization is available for dative-absolutive clauses supports our treatment of the dative argument as the subject.

Jirrbal. *Jirrbal* or *Dyirbal* (Pama-Nyungan, Australia) (Dixon 1972, 1994) is a famous case of a language that has been argued to have transitive clauses with an S and P pivot. It has a combination of ergative-absolutive case marking on nouns (similar to Basque), as in Example (71a) and Example (72a), and nominative-accusative case marking on pronouns, as in Example (71b) and Example (72b), a common pattern of split ergative case marking. In both cases, in transitive clauses, we treat the P pivot core argument as the *nsubj* and the A core argument as an *obj*, but we mark them for unusual semantic role alignment with *nsubj:pass* and *obj:agent*, respectively.

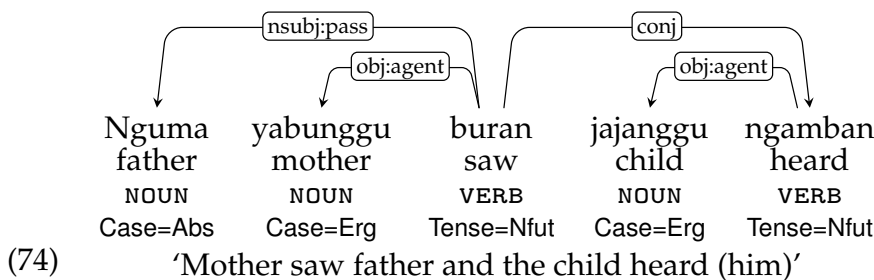




There are several grammatical processes, such as relativization, which, when restricted in application in a language, frequently only apply to subjects. The motivation for the above analysis is that in Jirrbal these processes apply to the S and P core arguments. For instance, the role of the head noun in a relative clause must be S or P, allowing relative clauses like Example (73a) where the relativized role is P, but not a relative clause where the relativized role is A. To express such an idea, the relative clause must be antipassivized, making the previous P into an oblique and the previous A into an S pivot, as in Example (73b).



As another example, the shared arguments in coordinated clauses must be S or P pivot core arguments, allowing the normally unexpected coordination in Example (74) but not allowing 'Mother saw father and heard the child' with a shared argument, except by antipassivization of the second clause. Again, this is most naturally handled by recognizing an S/P pivot which is analyzed as the nsubj in UD.



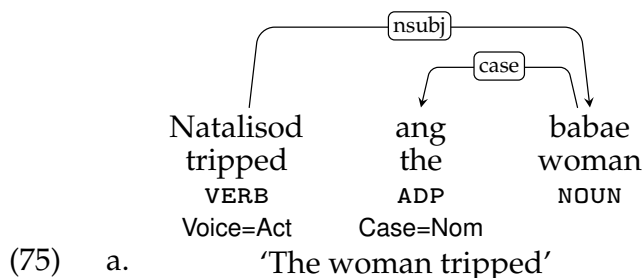
4.5 Other Voice Systems

In European languages, the contrast between the active and passive voice is an important factor in categorizing simple clauses and their arguments. Ergative languages sometimes have an analogous contrast between the active and the antipassive. Yet there are languages whose voice systems do not seem to fit easily into either of these patterns. In this subsection, we first look at Tagalog, a representative of the Philippine-type languages, which are sometimes subsumed in a larger group of **symmetrical voice languages** (Himmelmann 2005). Then, we will discuss the direct–inverse voice system of Algonquian languages, exemplified by Plains Cree.

Tagalog. The arguments in Tagalog are marked by function words that could be analyzed as either prepositions, or case-bearing determiners; the former analysis is adopted here.¹⁷ Although adpositions are often associated with oblique arguments and adjuncts, we have seen that it is not a universal rule. Spanish marks an animate direct object with the preposition *a*, and in Japanese all arguments are marked by postpositions, including the subject and the direct object.

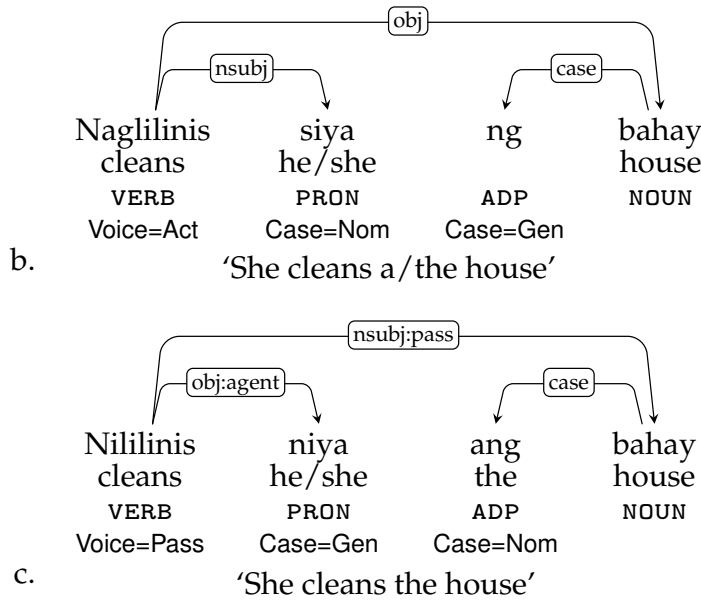
The most subject-like argument (again called the pivot) is marked by the preposition *ang*. Other core arguments (if any) are marked by the preposition *ng* (Kroeger 1993, pp. 40–47). A different set of prepositions is used with proper nouns. Personal pronouns are not used with prepositional markers but inflect for case. Verbs are marked with infix voice markers.

There is disagreement about whether the pivot is a subject and whether Tagalog has a subject at all. Andrews (2007, pp. 210–211) distinguishes two grammatical relations, the a-subject and the p-subject, each having some properties that are often associated with subjects in European languages. He also says that the actor “has subject-like properties regardless of whether or not it is the pivot.” For the purpose of easy and consistent annotation of UD, it is advantageous to follow the analysis of Manning (1996) and to always reserve the *nsubj* relation for the *ang*-phrase (the pivot), as in Example (75a). In the transitive sentences in Example (75b–c), different voices give different alignments of semantic roles to grammatical relations. We mark prepositions and personal pronouns with the Case feature: the pivot with nominative, and the other core argument(s) with genitive.¹⁸



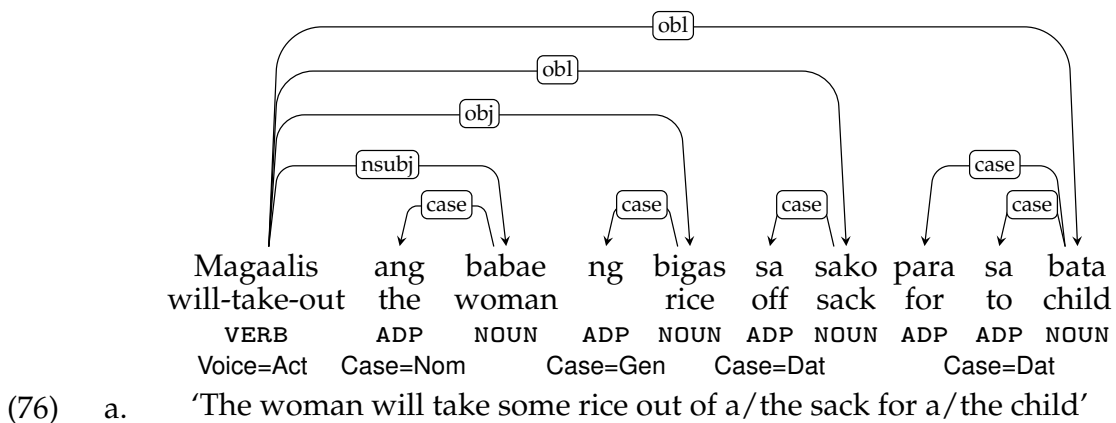
17 There is no standard terminology for these words in the literature. Some authors classify them as prepositions (e.g., Schachter and Shopen 2007, p. 35), some as articles or determiners (e.g., Dryer 2007, pp. 94–95 and 121–122), and many authors avoid either of the terms and use the term “markers” instead (e.g., Andrews 2007, p. 203).

18 The names for the cases are not without controversy either. If the subject is nominative, the other core argument could be expected to be accusative, but due to its other functions, Tagalog *ng* is often glossed as genitive (Himmelmann 2005). The nominative–accusative analysis has been advocated by some authors (e.g., Guilfoyle, Hung, and Travis 1992), while others prefer to analyze Tagalog as an ergative–absolutive language (e.g., Payne 1982; De Guzman 1988; Gerdtz 1988), which would mean that the pivot is in the absolutive and the *ng*-phrase in the ergative.

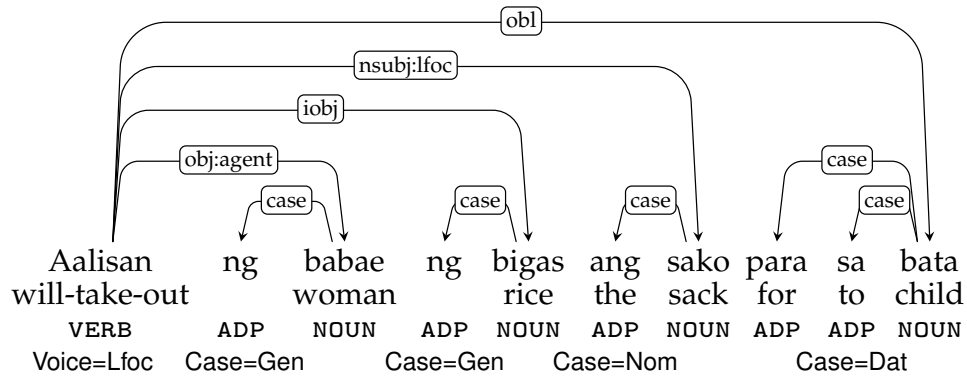


Despite the fact that we conveniently reuse the active and passive voice labels, it has to be understood that this alternation is significantly different from the active–passive alternation in English. Both clauses are transitive, as the non-subject argument stays core; in an English passive clause, the actor would be demoted to an oblique dependent. The construction in Example (75c) is neither less frequent nor morphosyntactically more complex than Example (75b). That is why the Austronesian voice system has been described as symmetrical; rather than “active” and “passive,” the voice labels should be read as “agent/actor-focus” and “patient/theme-focus,” respectively.

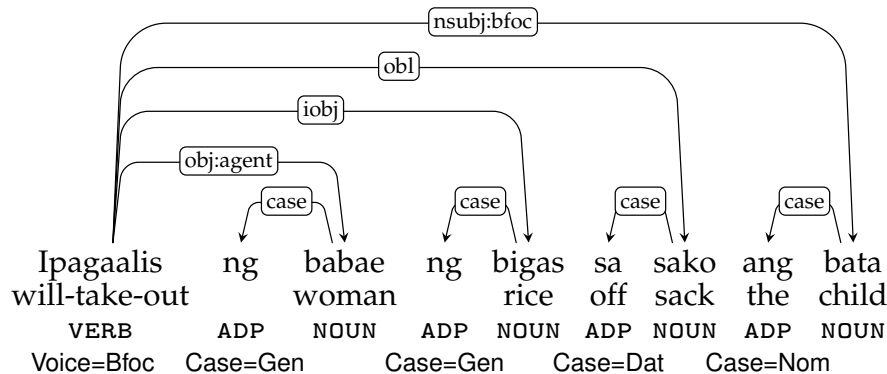
Locative, directional, and benefactive nominals are normally coded as oblique (e.g., the dative *sa sako* ‘from sack’ in Example (76a)). However, there are additional voices where these nominals become subjects, such as the location-focus voice in Example (76b) and the beneficiary-focus voice in Example (76c). One of the reasons why a dependent is promoted to the subject is that the subject is understood as the topic of the sentence.¹⁹



19 The “focus” in the names of the voices indicates that the verb “focuses” on a particular semantic role and it should not be confused with pragmatic focus, which is the opposite of “topic.”

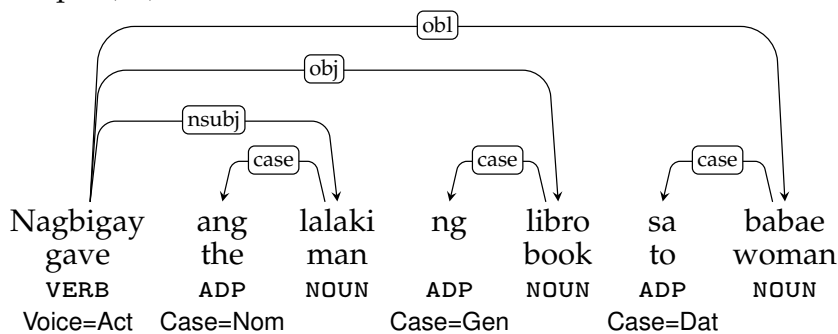


b. 'A/the woman will take some rice out of the sack for a/the child'



c. 'A/the woman will take some rice out of a/the sack for the child'

Because the agent and patient stay core arguments even in the locative and beneficiary voices, Example (76b) and Example (76c) are ditransitive clauses with three core arguments. In contrast, the verbs of giving, which are typical representatives of ditransitive predicates in other languages, form a standard transitive clause in the "active" and "passive" voices, with the recipient coded as a directional (dative) oblique dependent, as in Example (77).



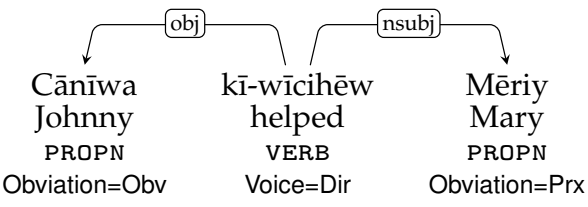
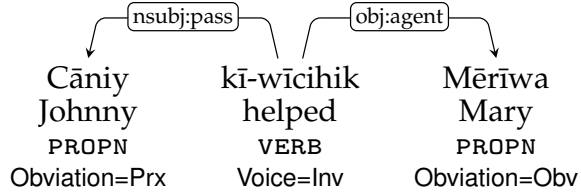
(77) 'The man gave a book to the woman'

Plains Cree. The Algonquian (North American) language Plains Cree (Wolvengrey 2011) cross-references one or two core arguments by verbal inflection, which is sufficient to allow for a relatively free word order. As in many other languages where person and number of an argument is cross-referenced by the verb, the argument does not need to appear overtly. The distinguishing feature of the verb forms in Example (78) is voice: Example (78a) is in the direct voice (Dir), where higher arguments in the obliqueness hierarchy are taken to be more agent-like, whereas Example (78b) is in the inverse voice (Inv), where lower arguments are taken to be more agent-like. Given that first person arguments are higher than third person arguments, the agent is 'we' and the patient is 'they' in Example (78a), and inversely in Example (78b).

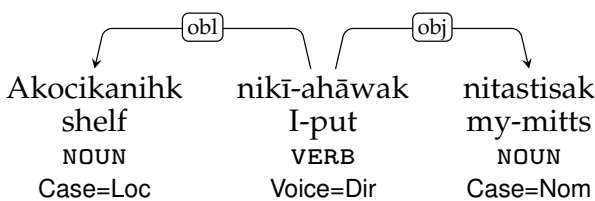
- (78) a. Niwīcihānānak
1Plur[high]-help-Dir-3[low]-Plur[low]
'We help them'
- b. Niwīcihikonānak
1Plur[high]-help-Inv-3[low]-Plur[low]
'They help us'

Arguments cross-referenced by the verb are without doubt core arguments. It is not so obvious how to label the two arguments, as Plains Cree does not clearly have a subject in the Indo-European sense. It is one of a number of languages where evidence for differentiating core grammatical relations except via semantic role seems limited or non-existent. Nevertheless, it seems best to postulate that the argument higher in the obliqueness hierarchy should get the label *nsubj* in UD; the other core argument then gets *obj*. Such a distinction can be annotated easily and consistently. The subject will be more agent-like in the direct voice, and more patient-like in the inverse voice. This can be signaled by labeling non-agentive subjects as *nsubj:pass* without explicitly claiming that such sentences are passivized, unlike Dahlstrom (1991).

If two animate third-person arguments are involved, one of them is considered *proximate* (more topical, higher in the obliqueness hierarchy) and the other is considered *obviative* (less topical, lower in the obliqueness hierarchy). The obviative noun is marked morphologically by the suffix *-a*. We define a language-specific morphological feature, *Obviation*, with the values *Prx* and *Obv*, to represent this. In Example (79a), *Mēriy* is proximate, hence it is the subject, and it is also the agent because the verb is in the direct voice. In Example (79b), *Cāniy* is proximate and thus the subject; however, *Mēriy* is still the agent because the verb is in the inverse voice.

- (79) a. 
'Mary helped Johnny'
- b. 
'Mary helped Johnny / Johnny was helped by Mary'

Even though Plains Cree does not use morphological cases to distinguish agents from patients, nouns have a locative case (*Case=Loc*) that marks the noun as oblique and unable to be cross-referenced by verbal inflection.

- (80) 
'I put my mitts on the shelf'

While much work remains to be done in descriptive linguistics and its implementation in UD, we hope that this survey of typologically different languages has shown that UD provides a workable framework for the description and annotation of a broad range of clause-marking choices.

5. Design Principles of UD

There are many different ways that UD could have been designed. In this section, we briefly motivate and explain the design principles that guided us. Importantly, what UD seeks to achieve is rather different to what a grammar formalism in theoretical linguistics typically seeks to achieve, and thus the outcome is quite different.

The overarching goal of UD is a crosslinguistically consistent universal grammar that is suitable for use by the common person. That is, UD should be informed by our linguistic knowledge and the typology of language variation, but it should be something simple and interpretable enough that a psychologist, a software engineer, or a high school English teacher can comfortably use it. Behind this goal is a belief that there is something in common between human languages to be captured; as Bresnan et al. (2016, p. 1) argues, “there must be ... a common organizing structure of all languages that underlies their superficial variations in modes of expression.” From a linguistic point of view, such a common organizing structure is necessary for comparative linguistic studies and a substantive theory of crosslinguistic typology. From a practical NLP viewpoint, a common framework is needed to make it easy to build and maintain multilingual NLP systems, to allow effective crosslinguistic transfer learning, to enable meaningful crosslinguistic comparisons of parsing difficulty, and to approach the goal of a universal parser that works for all languages based on modern universal neural encodings of text (see, e.g., Kondratyuk and Straka 2019).

In choosing a common organizing structure for human language, UD applies a version of the Goldilocks principle: We should aim to maximize the commonality between languages but not to an extent that it obscures genuine differences between languages. Seeking commonality, it is a mistake if a parallel morphosyntactic notion is unnecessarily annotated inconsistently across different languages. Seeking fidelity, we avoid annotating things that are actually different (such as morphological vs. periphrastic expression of tense) as if they were the same. As a special case, UD eschews annotating things that are not there (empty items), because this is usually an artificial device to increase parallelism. Practically, we deal with quirky features of particular languages by insisting on use of a universal taxonomy of categories, features, and relations, but allowing the use of language-specific elaboration via subcategories. While the pressure in theoretical linguistics is for representations to become more and more detailed and complex over time, for UD, we realize that often less is better.

The secret to understanding the design and success of UD is to realize that the design is a very subtle compromise between a number of competing criteria:

1. UD needs to be reasonably satisfactory on linguistic analysis grounds for individual languages—a journeyman’s universal grammar.
2. UD needs to be good for linguistic typology: It should bring out crosslinguistic parallelism across languages and language families.
3. UD must be suitable for rapid, consistent annotation by a human annotator.
4. UD must be easily comprehended and used by non-linguist users with prosaic needs.

5. UD must be suitable for computer parsing with high accuracy.
6. UD must support well downstream language understanding tasks, such as relation extraction, reading comprehension, machine translation, and so on.

We observe that it is very easy to come up with a proposal that improves UD on one of these dimensions. The interesting and difficult part of developing UD has been working to improve the scheme and annotation guidelines while remaining sensitive to all these dimensions. Compare the analogy that school children are taught that English has eight parts of speech: Noun, Verb, Adjective, Adverb, Pronoun, Preposition, Interjection, Conjunction. This is not really true, but it has enough fidelity, enough simplicity, and enough comprehensibility to satisfy most people.

Many of the high-level design decisions of UD can be motivated in terms of these criteria. Making UD a monostratal theory—a theory with one representation (cf. Ladusaw 1988)—facilitates easy annotation and parsing. The emphasis on grammatical relations works well for both comparative linguistics and usage by non-linguists. Preferring relations between content words rather than mediated by function words increases crosslinguistic parallelism and within language parallelism (simple vs. periphrastic tenses become more parallel), and makes relation extraction easier (fewer, smaller patterns will cover a broader range of data). For example, the construction of predicating a property of a nominal (*the sky is blue*) is universal, while the strategy of achieving this via an auxiliary or copula verb is not. We increase parallelism by having a dependency between the nominal and the predicate. It also has the effect of more perspicuously revealing predicate–argument structure to the benefit of downstream processing. By mainly adopting terminology from traditional (European) grammar, we make it easier for non-expert users to comprehend UD representations, but we still make some changes, such as using the term **adposition**, to make UD more satisfactory on cross-linguistic grounds.

A key choice was between dependency representations and constituency representations (also known as phrase structure grammar, context-free grammar, or immediate constituency representations). One motivation here was simply the direction of the field of computational linguistics. While the famous early treebanks of modern empirical NLP, the Lancaster/IBM Treebank (Black, Garside, and Leech 1993) and the Penn Treebank (Marcus, Santorini, and Marcinkiewicz 1993), and many treebanks that followed thereafter were constituency treebanks, by the early 2000s, there had been a huge shift to the use of dependency treebanks in computational linguistics. This was not altogether a new thing. David Hayes, a founder of the Association for Computational Linguistics, had strongly advocated for the use of Dependency Grammar in the 1960s (Hayes 1964). And it was not a random shift: The adoption of a dependency representation was driven by several of the ideas that underlie our design principles, such as simplicity, easy cross-linguistic applicability, interpretability by non-linguists, and usefulness for downstream applications.

Our goal was for UD to be a lightweight representation that is easy and satisfactory for people to work with. It is gratifying to see that many people from disparate linguistic and non-linguistic backgrounds have found UD congenial enough that they have felt able and motivated to use it.

6. Conclusion and Outlook

In this article, we have articulated the linguistic theory underlying the UD framework. After discussing basic theoretical assumptions (Section 2), we showed how the theory

applies to a wide range of linguistic constructions (Section 3), zoomed in on the treatment of core arguments in a diverse sample of languages (Section 4), and concluded by revisiting the design principles of UD (Section 5). We argued that UD provides a good foundation for crosslinguistically consistent morphosyntactic annotation, which can support research and application development in NLP, as well as typologically oriented studies in linguistics. The UD resources have already had a significant impact on NLP research, most notably for multilingual dependency parsing through two editions of CoNLL shared tasks (Zeman et al. 2017, 2018), which have created a new generation of parsers that handle a large number of languages and that parse from raw text rather than relying on pre-tokenized input. The resources have also been widely used for research on cross-lingual and polyglot parsing, as well as universal semantic parsing (see, e.g., Tiedemann 2015; Agić 2017; Kondratyuk and Straka 2019; Reddy et al. 2017), where the availability of resources with crosslinguistically consistent annotation is crucial. Among more linguistically oriented studies, we find research on psycholinguistics and especially word order typology (see, e.g., Futrell, Mahowald, and Gibson 2015; Naranjo and Becker 2018; Levshina 2019). For an overview of UD-related research, we refer to the proceedings from the annual UD workshops (de Marneffe, Nivre, and Schuster 2017; de Marneffe, Lynn, and Schuster 2018; Rademaker and Tyers 2019; de Marneffe et al. 2020).

Before we conclude, it is important to note that there are many details of the theory that still need to be worked out. Even though all major construction types are covered by the current version of the UD guidelines, there are many specific phenomena and special cases that have not been discussed in sufficient detail or received a definitive treatment in UD. Moreover, the list of such phenomena constantly grows as new languages are considered for analysis in the UD framework. Therefore, while we regard the core of the UD theory as stable, we expect the theory as a whole to continue to evolve over time, as a result of the ongoing dialogue between experts on different languages trying to find the right balance between language-specific and universal perspectives in the application of UD to their language. We look forward to continuing that dialogue and welcome everyone who is interested to take part in it.

Acknowledgments

Many people have contributed to the development of UD, and we especially want to mention our colleagues in the UD core guidelines group, Filip Ginter, Yoav Goldberg, Jan Hajič, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Sebastian Schuster, Natalia Silveira, Reut Tsarfaty, and Francis Tyers, as well as William Croft, Kim Gerdes, Sylvain Kahane, Nathan Schneider, and Amir Zeldes. We are grateful to Google for sponsoring the UD project in a number of ways, and to the *Computational Linguistics* reviewers for helpful suggestions. Daniel Zeman's and Joakim Nivre's contributions to this work were supported by grant GX20-16819X of the Czech Science Foundation and grant 2016-01817 of the Swedish Research Council, respectively.

References

- Agić, Željko. 2017. Cross-lingual parser selection for low-resource languages. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 1–10, Göteborg.
- Andrews, Avery D. 2007. The major functions of the noun phrase. In Timothy Shopen, editor, *Language Typology and Syntactic Description. Volume I: Clause Structure*, Cambridge University Press, pages 132–223. <https://doi.org/10.1017/CB09780511619427.003>
- Aronoff, Mark. 2007. In the beginning was the word. *Language*, 83:803–830. <https://doi.org/10.1353/lan.2008.0042>
- Badawi, Elsaid, M. G. Carter, and Adrian Gully. 2013. *Modern Written Arabic: A Comprehensive Grammar*. Routledge, London/New York. <https://doi.org/10.4324/9780203351758>

- Baker, Collin F., Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal. <https://doi.org/10.3115/980845.980860>
- Black, Ezra, Roger Garside, and Geoffrey Leech, editors. 1993. *Statistically-Driven Computer Grammars of English: The IBM/Lancaster Approach*. Rodopi, Amsterdam.
- Blevins, James P. 2006. Word-based morphology. *Journal of Linguistics*, 42:531–573. <https://doi.org/10.1017/S0022226706004191>
- Blevins, James P., Farrell Ackerman, and Robert Malouf. 2017. Word and paradigm morphology. In Jenny Audring and Francesca Masini, editors, *The Oxford Handbook of Morphological Theory*. Oxford University Press, Oxford, pages 265–284. <https://doi.org/10.1093/oxfordhb/9780199668984.013.22>
- Boneh, Nora and Léa Nash. 2012. Core and non-core datives in French. In Beatriz Fernández and Ricardo Etxepare, editors, *Variation in Datives*. Oxford University Press, Oxford. pages 22–49. <https://doi.org/10.1093/acprof:oso/9780199937363.003.0002>
- Bouma, Gosse, Jan Hajič, Dag Haug, Joakim Nivre, Per Erik Solberg, and Lilja Øvrelid. 2018. Expletives in Universal Dependency Treebanks. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 18–26, Bruxelles. <https://doi.org/10.18653/v1/W18-6003>
- Bresnan, Joan, Ash Asudeh, Ida Toivonen, and Stephen Wechsler. 2016. *Lexical-Functional Syntax*, 2nd edition. Wiley-Blackwell, Chichester. <https://doi.org/10.1002/9781119105664>
- Bresnan, Joan and Sam A. Mchombo. 1995. The lexical integrity principle: Evidence from Bantu. *Natural Language and Linguistic Theory*, 13:181–254. <https://doi.org/10.1007/BF00992782>
- Chomsky, Noam. 1970. Remarks on nominalization. In Roderick A. Jacobs and Peter S. Rosenbaum, editors, *Readings in English Transformational Grammar*. Ginn and Co., pages 11–61.
- Comrie, Bernhard. 1981. *Language Universals and Linguistic Typology: Syntax and Morphology*. Basil Blackwell, Oxford.
- Croft, William. 1991. *Syntactic Categories and Grammatical Relations: The Cognitive Organization of Information*. University of Chicago Press.
- Croft, William. 2001. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford University Press. <https://doi.org/10.1093/acprof:oso/9780198299554.001.0001>
- Croft, William. 2002. *Typology and Universals*, second edition, Cambridge University Press.
- Croft, William. forthcoming. *Morphosyntax: Constructions of the World's Languages*. Cambridge University Press, Cambridge.
- Dahlstrom, Amy. 1991. *Plains Cree Morphosyntax*. Garland, New York.
- Dalrymple, Mary. 2001. *Lexical-Functional Grammar*. Academic Press. <https://doi.org/10.1163/9781849500104>
- De Guzman, Videia. 1988. Ergative analysis for Philippine languages: An analysis. In Richard McGinn, editor, *Studies in Austronesian Linguistics*. Ohio University Center for International Studies, Athens, OH, pages 323–345.
- de Marneffe, Marie Catherine, Miryam de Lhoneux, Joakim Nivre, and Sebastian Schuster, editors. 2020. *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*. Online. <https://www.aclweb.org/anthology/2020.udw-1.0>
- de Marneffe, Marie Catherine, Teresa Lynn, and Sebastian Schuster, editors. 2018. *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*. Bruxelles.
- de Marneffe, Marie Catherine, Joakim Nivre, and Sebastian Schuster, editors. 2017. *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*. Göteborg.
- Dione, Cheikh Bamba. 2019. Developing Universal Dependencies for Wolof. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 12–23, Paris. <https://doi.org/10.18653/v1/W19-8003>
- Dixon, R. M. W. 1972. *The Dyirbal Language of North Queensland*. Cambridge University Press, Cambridge. <https://doi.org/10.1017/CB09781139084987>
- Dixon, R. M. W. 1994. *Ergativity*. Cambridge University Press, Cambridge.
- Dixon, R. M. W. 2009. *Basic Linguistic Theory. Volume 1: Methodology*. Oxford University Press.
- Dryer, Matthew S. 2007. Word order. In Timothy Shopen, editor, *Language Typology and Syntactic Description. Volume I: Clause Structure*, second edition. Cambridge

- University Press, Cambridge, pages 61–131. <https://doi.org/10.1017/CB09780511619427.002>
- Farrar, Scott and D. Terence Langendoen. 2003. A linguistic ontology for the semantic Web. *GLOT International*, 7:97–100.
- Fischer, Wolfdietrich. 1997. Classical Arabic. In Robert Hetzron, editor, *The Semitic Languages*, Routledge, London/New York, pages 187–219.
- Foley, William A. 2007. A typology of information packaging in the clause. In Timothy Shopen, editor, *Language Typology and Syntactic Description. Volume I: Clause Structure*, second edition. Cambridge University Press, Cambridge, pages 362–446. <https://doi.org/10.1017/CB09780511619427.007>
- Futrell, Richard, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341. <https://doi.org/10.1073/pnas.1502134112>, PubMed: 26240370
- Gerdes, Kim, 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, Bruxelles. <https://doi.org/10.18653/v1/W18-6008>
- Gerdes, Kim and Sylvain Kahane. 2016. Dependency annotation choices: Assessing theoretical and practical issues of universal dependencies. In *Proceedings of LAW X – The 10th Linguistic Annotation Workshop*, pages 131–140, Berlin. <https://doi.org/10.18653/v1/W16-1715>
- Gerds, Donna. 1988. Antipassives and causatives in Ilokano: Evidence for an ergative analysis. In Richard McGinn, editor, *Studies in Austronesian Linguistics*, Ohio University Center for International Studies, Athens, OH, pages 323–345.
- Greenberg, Joseph H. 1963. Some universals of grammar with particular reference to the order of meaningful elements. In Joseph H. Greenberg, editor, *Universals of Human Language*, MIT Press, pages 73–113.
- Grimshaw, Jane. 1991 [2005]. Extended projection. Ms., Brandeis University. Appears in Jane Grimshaw (2005), *Words and Structure*. Stanford, CA: CSLI Publications, pages 1–74.
- Guilfoyle, Eichne, Henrietta Hung, and Lisa Travis. 1992. Spec of IP and Spec of VP: Two subjects in Austronesian languages. *Natural Language and Linguistic Theory*, 10:375–414. <https://doi.org/10.1007/BF00133368>
- Haspelmath, Martin. 2001. Word classes and parts of speech. In *International Encyclopedia of the Social & Behavioral Sciences*. Elsevier Science, pages 16538–16545. <https://doi.org/10.1016/B0-08-043076-7/02959-4>
- Haspelmath, Martin. 2011a. The indeterminacy of word segmentation and the nature of morphology and syntax. *Folia Linguistica*, 45:31–80. <https://doi.org/10.1515/flin.2011.002>
- Haspelmath, Martin. 2011b. On S, A, P, T, and R as comparative concepts for alignment typology. *Linguistic Typology*, 15:535–567. <https://doi.org/10.1515/LITY.2011.035>
- Haspelmath, Martin. 2014. Arguments and adjuncts as language-particular syntactic categories and as comparative concepts. *Linguistic Discovery*, 12(2):3–11. <https://doi.org/10.1349/PS1.1537-0852.A.442>
- Haspelmath, Martin. 2015. Ditransitive constructions. *Annual Review of Linguistics*, 1:19–41. <https://doi.org/10.1146/annurev-linguist-030514-125204>
- Haspelmath, Martin. 2019. Indexing and flagging, and head and dependent marking. *Te Reo*, 62(1):93–115.
- Hayes, David G. 1964. Dependency theory: A formalism and some observations. *Language*, 40(4):511–525. <https://doi.org/10.2307/411934>
- Himmelmann, Nikolaus P. 2005. The Austronesian languages of Asia and Madagascar: Typological characteristics. In Alexander Adelaar and Nikolaus P. Himmelmann, editors, *The Austronesian Languages of Asia and Madagascar*. Routledge, London/New York, pages 110–181.
- Hopper, Paul J. and Elizabeth Traugott. 2003. *Grammaticalization*. Cambridge University Press. <https://doi.org/10.1017/CB09781139165525>
- Huddleston, Rodney and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press. <https://doi.org/10.1017/9781316423530>
- Hudson, Richard A. 1984. *Word Grammar*. Blackwell.
- Hudson, Richard A. 1990. *English Word Grammar*. Blackwell.
- Kaplan, Ron and Joan Bresnan. 1982. *Lexical-Functional Grammar: A formal*

- system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, MIT Press, pages 173–281.
- Kayne, Richard S. 1984. *Connectedness and Binary Branching*. Foris Publications, Dordrecht. <https://doi.org/10.1515/9783111682228>
- Kondratyuk, Daniel and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2779–2795, Hong Kong. <https://doi.org/10.18653/v1/D19-1279>
- Kroeger, Paul R. 1993. *Phrase Structure and Grammatical Relations in Tagalog*. Stanford University Press, Stanford, CA, USA.
- Ladusaw, William A. 1988. A proposed distinction between levels and strata: In *Linguistics in the Morning Calm 2: Selected Papers from SICOL-1986*. The Linguistic Society of Korea, Hanshin, Seoul.
- Levshina, Natalia. 2019. Token-based typology and word order entropy: A study based on universal dependencies. *Linguistic Typology*, 23(3):533–572. <https://doi.org/10.1515/lingty-2019-0025>
- Manning, Christopher D. 1996. *Ergativity: Argument Structure and Grammatical Relations*. CSLI Publications, Stanford, CA.
- Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330. <https://doi.org/10.21236/ADA273556>
- Mel'čuk, Igor. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press.
- Milicevic, Jasmina. 2006. A short guide to the Meaning-Text linguistic theory. *Journal of Koralex*, 8:187–233.
- Naranjo, Matías Guzmán and Laura Becker. 2018. Quantitative word order typology with UD. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)*, pages 91–104, Oslo.
- Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Dan Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, Portorož.
- Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2020)*, pages 4027–4036. Online.
- Osborne, Timothy and Kim Gerdes. 2019. The status of function words in dependency grammar: A critique of Universal Dependencies (UD). *Glossa*, 4(1):17.1–28. <https://doi.org/10.5334/gjgl.537>
- Oyharçabal, Bernard. 2003. Lexical causatives and causative alternation in Basque. In Bernard Oyharçabal, editor, *Inquiries into the Syntax-Lexicon relations in Basque*, number XLVI in *Supplements of ASJU*. Euskal Herriko Unibertsitatea, pages 223–253.
- Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank. *Computational Linguistics*, 31:71–106. <https://doi.org/10.1162/0891201053630264>
- Payne, Thomas. 1982. Role and reference related subject properties and ergativity in Yup'ik Eskimo and Tagalog. *Studies in Language*, 6:75–106. <https://doi.org/10.1075/sl.6.1.05pay>
- Perlmutter, David M., editor. 1983. *Studies in Relational Grammar*. The University of Chicago Press.
- Pineda, Anna. 2013. Romance double object constructions and transitivity alternations. In *Proceedings of ConSOLE XX*, pages 185–211, Leipzig.
- Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. CSLI Publications.
- Przepiórkowski, Adam. 2016. How Not to distinguish arguments from adjuncts in LFG. In *Proceedings of the Joint 2016 Conference on Head-Driven Phrase Structure Grammar and Lexical Functional Grammar*, pages 560–580, Warszawa.
- Rademaker, Alexandre and Francis Tyers, editors. 2019. *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*. Paris.
- Reddy, Siva, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*,

- pages 89–101, Copenhagen. <https://doi.org/10.18653/v1/D17-1009>
- Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proceedings of CICLing*, pages 1–15, Mexico City. https://doi.org/10.1007/3-540-45715-1_1
- Schachter, Paul and Timothy Shopen. 2007. Part-of-speech systems. In Timothy Shopen, editor, *Language Typology and Syntactic Description. Volume I: Clause Structure*, Cambridge University Press, second edition. Cambridge, pages 1–60. <https://doi.org/10.1017/CB09780511619427.001>
- Sgall, Petr, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Pragmatic Aspects*. Reidel.
- Spencer, Andrew and Ana R. Luís. 2012. *Clitics: An Introduction*. Cambridge University Press, Cambridge.
- Stump, Gregory T. 2001. *Inflectional Morphology: A Theory of Paradigm Structure*. Cambridge University Press. <https://doi.org/10.1017/CB09780511486333>
- Sylak-Glassman, John, Christo Kirov, David Yarowsky, and Roger Que. 2015. A language-independent feature schema for inflectional morphology. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 674–680, Beijing. <https://doi.org/10.3115/v1/P15-2111>
- Tesnière, Lucien. 1959. *Éléments de syntaxe structurale*. Editions Klincksieck, Paris.
- Tesnière, Lucien. 2015 [1959]. *Elements of Structural Syntax*. Translation by Timothy Osborne and Sylvain Kahane of Tesnière (1959). John Benjamins. <https://doi.org/10.1075/z.185>
- Thompson, Sandra A. 1997. Discourse motivations for the core-oblique distinction as a language universal. In Akio Kamio, editor, *Directions in Functional Linguistics*. John Benjamins, pages 59–82. <https://doi.org/10.1075/slcs.36.06tho>
- Tiedemann, Jörg. 2015. Cross-lingual dependency parsing with Universal Dependencies and predicted PoS labels. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling)*, pages 340–349, Uppsala.
- Van Peteghem, Marleen. 2006. Le datif en français: un cas structural. *Journal of French Language Studies*, 16:93–110. <https://doi.org/10.1017/S0959269506002286>
- Van Valin, Jr., Robert D., editor. 1993. *Advances in Role and Reference Grammar*. John Benjamins. <https://doi.org/10.1075/cilt.82>
- Whorf, Benjamin Lee. 1956. *Language, Thought, and Reality*. MIT Press.
- Wolwengrey, Arok Elessar. 2011. *Semantic and pragmatic functions in Plains Cree syntax*. Ph.D. thesis, LOT, Utrecht, Netherlands.
- Zeman, Daniel, 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, Bruxelles.
- Zeman, Daniel, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gökırmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, et al. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver. <https://doi.org/10.18653/v1/K17-3001>
- Zwicky, Arnold M. and Geoffrey K. Pullum. 1983. Cliticization vs. inflection: English *n't*. *Language*, 59:502–513. <https://doi.org/10.2307/413900>
- Zúñiga, Fernando and Beatriz Fernández. 2019. Grammatical relations in Basque. In Balthasar Bickel and Alena Witzlack-Makarevich, editors, *Argument selectors: A new perspective on grammatical relations*, Typological Studies in Language, volume 123 of *Typological Studies in Language*. John Benjamins, Amsterdam, pages 185–211. <https://doi.org/10.1075/tsl.123.06zun>

6.5 CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies

Full reference: Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 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, Bruxelles, Belgium, October 2018. Association for Computational Linguistics. DOI 10.18653/v1/K18-2001. URL <https://aclanthology.org/K18-2001v1.pdf>. [Zeman et al., 2018]

Comments: The overview paper of the second UD shared task in 2018 is presented here as a culmination of the two-year long evaluation campaign (Chapter 4); the first task was described in Zeman et al. [2017]. Five years later this paper remains an important reference for multilingual end-to-end parsing, although new and better parsing models have emerged since then, especially with the advent of transformer-based multilingual large language models. We also organized two more shared tasks collocated with the IWPT conference [Bouma et al., 2020, 2021], which were focused on Enhanced UD parsing (Chapter 5) but all the previous annotation levels were evaluated as well. Unlike the pre-UD parsing tasks, new parsers are usually not evaluated on the shared task data except for comparison purposes; instead, they are evaluated on the most recent release of UD, which includes new languages and potentially also fixes of annotation errors in the older datasets. End-to-end parsing evaluation has become standard, and the shared task evaluation script is freely available among UD tools so that everyone can evaluate their parser following the same methodology. As for the newly proposed evaluation metrics, they cannot compete in popularity with the well-established LAS, yet they are occasionally used by other authors (e.g., Dary and Nasr [2021]). My contribution: about 45%. Number of citations according to Google Scholar (retrieved 2023-07-21): **569**.²

²Google Scholar has merged the two papers about the two shared task years. This is the aggregate number of citations for both [Zeman et al., 2017] and [Zeman et al., 2018].

CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies

Daniel Zeman¹, Jan Hajič¹, Martin Popel¹, Martin Potthast²,
Milan Straka¹, Filip Ginter³, Joakim Nivre⁴, and Slav Petrov⁵

¹Charles University, Faculty of Mathematics and Physics

²Universität Leipzig, ³University of Turku

⁴Uppsala University, ⁵Google AI Language

{zeman|hajic|popel|straka}@ufal.mff.cuni.cz,
martin.potthast@uni-leipzig.de, figint@utu.fi,
joakim.nivre@lingfil.uu.se, slav@google.com

Abstract

Every year, the Conference on Computational Natural Language Learning (CoNLL) features a shared task, in which participants train and test their learning systems on the same data sets. In 2018, one of two tasks was devoted to learning dependency parsers for a large number of languages, in a real-world setting without any gold-standard annotation on the input. All test sets followed the unified annotation scheme of Universal Dependencies (Nivre et al., 2016). This shared task constitutes a 2nd edition—the first one took place in 2017 (Zeman et al., 2017); the main metric from 2017 was kept, allowing for easy comparison, and two new main metrics were introduced. New datasets added to the Universal Dependencies collection between mid-2017 and the spring of 2018 contributed to the increased difficulty of the task this year. In this overview paper, we define the task and the updated evaluation methodology, describe data preparation, report and analyze the main results, and provide a brief categorization of the different approaches of the participating systems.

1 Introduction

The 2017 CoNLL shared task on universal dependency parsing (Zeman et al., 2017) picked up the thread from the influential shared tasks in 2006

and 2007 (Buchholz and Marsi, 2006; Nivre et al., 2007) and evolved it in two ways: (1) the parsing process started from raw text rather than gold standard tokenization and part-of-speech tagging, and (2) the syntactic representations were consistent across languages thanks to the Universal Dependencies framework (Nivre et al., 2016). The 2018 CoNLL shared task on universal dependency parsing starts from the same premises but adds a focus on morphological analysis as well as data from new languages.

Like last year, participating systems minimally had to find labeled syntactic dependencies between words, i.e., a syntactic head for each word, and a label classifying the type of the dependency relation. In addition, this year’s task featured new metrics that also scored a system’s capacity to predict a morphological analysis of each word, including a part-of-speech tag, morphological features, and a lemma. Regardless of metric, the assumption was that the input should be raw text, with no gold-standard word or sentence segmentation, and no gold-standard morphological annotation. However, for teams who wanted to concentrate on one or more subtasks, segmentation and morphology predicted by the baseline UDPipe system (Straka et al., 2016) was made available just like last year.

There are eight new languages this year: Afrikaans, Armenian, Breton, Faroese, Naija, Old French, Serbian, and Thai; see Section 2 for more details. The two new evaluation metrics are described in Section 3.

2 Data

In general, we wanted the participating systems to be able to use any data that is available free of charge for research and educational purposes (so that follow-up research is not obstructed). We deliberately did not place upper bounds on data sizes (in contrast to e.g. [Nivre et al. \(2007\)](#)), despite the fact that processing large amounts of data may be difficult for some teams. Our primary objective was to determine the capability of current parsers provided with large amounts of freely available data.

In practice, the task was formally closed, i.e., we listed the approved data resources so that all participants were aware of their options. However, the selection was rather broad, ranging from Wikipedia dumps over the OPUS parallel corpora ([Tiedemann, 2012](#)) to morphological transducers. Some of the resources were proposed by the participating teams.

We provided dependency-annotated training and test data, and also large quantities of crawled raw texts. Other language resources are available from third-party servers and we only referred to the respective download sites.

2.1 Training Data: UD 2.2

Training and development data came from the Universal Dependencies (UD) 2.2 collection ([Nivre et al., 2018](#)). This year, the official UD release immediately followed the test phase of the shared task. The training and development data were available to the participating teams as a pre-release; these treebanks were then released exactly in the state in which they appeared in the task.¹ The participants were instructed to only use the UD data from the package released for the shared task. In theory, they could locate the (yet unreleased) test data in the development repositories on GitHub, but they were trusted that they would not attempt to do so.

82 UD treebanks in 57 languages were included in the shared task;² however, nine of the smaller treebanks consisted solely of test data, with no data at all or just a few sentences available for training. 16 languages had two or more treebanks

from different sources, often also from different domains.³ See Table 1 for an overview.

61 treebanks contain designated development data. Participants were asked not to use it for training proper but only for evaluation, development, tuning hyperparameters, doing error analysis etc. Seven treebanks have reasonably-sized training data but no development data; only two of them, Irish and North Sámi, are the sole treebanks of their respective languages. For those treebanks cross-validation had to be used during development, but the entire dataset could be used for training once hyperparameters were determined. Five treebanks consist of extra test sets: they have no training or development data of their own, but large training data exist in other treebanks of the same languages (Czech-PUD, English-PUD, Finnish-PUD, Japanese-Modern and Swedish-PUD, respectively). The remaining nine treebanks are low-resource languages. Their “training data” was either a tiny sample of a few dozen sentences (Armenian, Buryat, Kazakh, Kurmanji, Upper Sorbian), or there was no training data at all (Breton, Faroese, Naija, Thai). Unlike in the 2017 task, these languages were not “surprise languages”, that is, the participants knew well in advance what languages to expect. The last two languages are particularly difficult: Naija is a pidgin spoken in Nigeria; while it can be expected to bear some similarity to English, its spelling is significantly different from standard English, and no resources were available to learn it. Even harder was Thai with a writing system that does not separate words by spaces; the Facebook word vectors were probably the only resource among the approved additional data where participants could learn something about words in Thai ([Rosa and Mareček, 2018](#); [Smith et al., 2018](#)). It was also possible to exploit the fact that there is a 1-1 sentence mapping between the Thai test set and the other four PUD test sets.⁴

Participants received the training and development data with gold-standard tokenization, sentence segmentation, POS tags and dependency re-

¹UD 2.2 also contains other treebanks that were not included in the task for various reasons, and that may have been further developed even during the duration of the task.

²Compare with the 81 treebanks and 49 languages in the 2017 task.

³We distinguish treebanks of the same language by their short names or acronyms. Hence, the two treebanks of Ancient Greek are identified as Perseus and PROIEL, the three treebanks of Latin are ITTB, Perseus and PROIEL, etc.

⁴While the test datasets were not available to the teams when they developed their systems, the documentation of the treebanks was supplied together with the training data, hence the teams could learn that the PUD treebanks were parallel.

Language	Tbk Code	2017	TrWrds
Afrikaans	af_afribooms	NA	34 K
Ancient Greek	grc_perseus	grc	160 K
Ancient Greek	grc_proiel	grc_proiel	187 K
Arabic	ar_padt	ar	224 K
Armenian	hy_armtdp	NA	1 K
Basque	eu_bdt	eu	73 K
Breton	br_keb	NA	0 K
Bulgarian	bg_btb	bg	124 K
Buryat	bxr_bdt	bxr	0 K
Catalan	ca_ancora	ca	418 K
Chinese	zh_gsd	zh	97 K
Croatian	hr_set	hr	154 K
Czech	cs_cac	cs_cac	473 K
Czech	cs_fictree	NA	134 K
Czech	cs_pdt	cs	1,173 K
Czech	cs_pud	cs_pud	0 K
Danish	da_ddt	da	80 K
Dutch	nl_alpino	nl	186 K
Dutch	nl_lassysmall	nl_lassysmall	75 K
English	en_ewt	en	205 K
English	en_gum	NA	54 K
English	en_lines	en_lines	50 K
English	en_pud	en_pud	0 K
Estonian	et_edt	et	288 K
Faroese	fo_ofst	NA	0 K
Finnish	fi_ftb	fi_ftb	128 K
Finnish	fi_pud	fi_pud	0 K
Finnish	fi_tdt	fi	163 K
French	fr_gsd	fr	357 K
French	fr_sequoia	fr_sequoia	51 K
French	fr_spoken	NA	15 K
Galician	gl_ctg	gl	79 K
Galician	gl_treegal	gl_treegal	15 K
German	de_gsd	de	264 K
Gothic	got_proiel	got	35 K
Greek	el_gdt	el	42 K
Hebrew	he_htb	he	138 K
Hindi	hi_hdtb	hi	281 K
Hungarian	hu_szeged	hu	20 K
Indonesian	id_gsd	id	98 K
Irish	ga_idt	ga	14 K

Language	Tbk Code	2017	TrWrds
Italian	it_isdt	it	276 K
Italian	it_postwita	NA	99 K
Japanese	ja_gsd	ja	162 K
Japanese	ja_modern	NA	0 K
Kazakh	kk_ktb	kk	1 K
Korean	ko_gsd	ko	57 K
Korean	ko_kaist	NA	296 K
Kurmanji	kmr_mg	kmr	0 K
Latin	la_ittb	la_ittb	270 K
Latin	la_perseus	la	18 K
Latin	la_proiel	la_proiel	172 K
Latvian	lv_lvtb	lv	81 K
Naija	pcm_nsc	NA	0 K
North Sámi	sme_giella	sme	17 K
Norwegian	no_bokmaal	no_bokmaal	244 K
Norwegian	no_nynorsk	no_nynorsk	245 K
Norwegian	no_nynorskli	NA	4 K
Old Church Slavonic	cu_proiel	cu	37 K
Old French	fro_srcmf	NA	136 K
Persian	fa_seraji	fa	121 K
Polish	pl_lfg	NA	105 K
Polish	pl_sz	pl	63 K
Portuguese	pt_bosque	pt	207 K
Romanian	ro_rrt	ro	185 K
Russian	ru_syntagrus	ru_syntagrus	872 K
Russian	ru_taiga	NA	10 K
Serbian	sr_set	NA	66 K
Slovak	sk_snk	sk	81 K
Slovenian	sl_ssj	sl	113 K
Slovenian	sl_sst	sl_sst	19 K
Spanish	es_ancora	es_ancora	445 K
Swedish	sv_lines	sv_lines	48 K
Swedish	sv_pud	sv_pud	0 K
Swedish	sv_talbanken	sv	67 K
Thai	th_pud	NA	0 K
Turkish	tr_imst	tr	38 K
Ukrainian	uk_iu	uk	75 K
Upper Sorbian	hsb_ufal	hsb	0 K
Urdu	ur_udtb	ur	109 K
Uyghur	ug_udt	ug	19 K
Vietnamese	vi_vtb	vi	20 K

Table 1: Overview of the 82 test treebanks. **TbkCode** = Treebank identifier, consisting of the ISO 639 language code followed by a treebank-specific code. **2017** = Code of the corresponding treebank in the 2017 task if applicable (“NA” otherwise). **TrWrds** = Size of training data, rounded to the nearest thousand words.

lations; and for most languages also lemmas and morphological features.

Cross-domain and cross-language training was allowed and encouraged. Participants were free to train models on any combination of the training treebanks and apply it to any test set.

2.2 Supporting Data

To enable the induction of custom embeddings and the use of semi-supervised methods in general, the participants were provided with supporting resources primarily consisting of large text corpora for many languages in the task, as well as embeddings pre-trained on these corpora. In total, 5.9 M

sentences and 90 G words in 45 languages are available in CoNLL-U format (Ginter et al., 2017); the per-language sizes of the corpus are listed in Table 2.

See Zeman et al. (2017) for more details on how the raw texts and embeddings were processed. Note that the resource was originally prepared for the 2017 task and it was not extended to include the eight new languages; however, some of the new languages are covered by the word vectors provided by Facebook (Bojanowski et al., 2016) and approved for the shared task.

Language	Words
English (en)	9,441 M
German (de)	6,003 M
Portuguese (pt)	5,900 M
Spanish (es)	5,721 M
French (fr)	5,242 M
Polish (pl)	5,208 M
Indonesian (id)	5,205 M
Japanese (ja)	5,179 M
Italian (it)	5,136 M
Vietnamese (vi)	4,066 M
Turkish (tr)	3,477 M
Russian (ru)	3,201 M
Swedish (sv)	2,932 M
Dutch (nl)	2,914 M
Romanian (ro)	2,776 M
Czech (cs)	2,005 M
Hungarian (hu)	1,624 M
Danish (da)	1,564 M
Chinese (zh)	1,530 M
Norwegian-Bokmål (no)	1,305 M
Persian (fa)	1,120 M
Finnish (fi)	1,008 M
Arabic (ar)	963 M
Catalan (ca)	860 M
Slovak (sk)	811 M
Greek (el)	731 M
Hebrew (he)	615 M
Croatian (hr)	583 M
Ukrainian (uk)	538 M
Korean (ko)	527 M
Slovenian (sl)	522 M
Bulgarian (bg)	370 M
Estonian (et)	328 M
Latvian (lv)	276 M
Galician (gl)	262 M
Latin (la)	244 M
Basque (eu)	155 M
Hindi (hi)	91 M
Norwegian-Nynorsk (no)	76 M
Kazakh (kk)	54 M
Urdu (ur)	46 M
Irish (ga)	24 M
Ancient Greek (grc)	7 M
Uyghur (ug)	3 M
Kurdish (kmr)	3 M
Upper Sorbian (hsb)	2 M
Buryat (bxr)	413 K
North Sámi (sme)	331 K
Old Church Slavonic (cu)	28 K
Total	90,669 M

Table 2: Supporting data overview: Number of words (M = million; K = thousand) for each language.

2.3 Test Data: UD 2.2

Each of the 82 treebanks mentioned in Section 2.1 has a test set. Test sets from two different treebanks of one language were evaluated separately as if they were different languages. Every test set contains at least 10,000 words (including punctuation marks). UD 2.2 treebanks that were smaller than 10,000 words were excluded from the shared task. There was no upper limit on the test data; the largest treebank had a test set comprising 170K words. The test sets were officially released as a part of UD 2.2 immediately after the shared task.⁵

3 Evaluation Metrics

There are three main evaluation scores, dubbed **LAS**, **MLAS** and **BLEX**. All three metrics reflect word segmentation and relations between content words. **LAS** is identical to the main metric of the 2017 task, allowing for easy comparison; the other two metrics include part-of-speech tags, morphological features and lemmas. Participants who wanted to decrease task complexity could concentrate on improvements in just one metric; however, all systems were evaluated with all three metrics, and participants were strongly encouraged to output all relevant annotation, even if they just copy values predicted by the baseline model.

When parsers are applied to raw text, the metric must be adjusted to the possibility that the number of nodes in gold-standard annotation and in the system output vary. Therefore, the evaluation starts with aligning system nodes and gold nodes. A dependency relation cannot be counted as correct if one of the nodes could not be aligned to a gold node. See Section 3.4 and onward for more details on alignment.

The evaluation software is a Python script that computes the three main metrics and a number of additional statistics. It is freely available for download from the shared task website.⁶

3.1 LAS: Labeled Attachment Score

The standard evaluation metric of dependency parsing is the *labeled attachment score* (**LAS**), i.e., the percentage of nodes with correctly assigned reference to the parent node, including the label (type) of the relation. For scoring purposes, only

⁵<http://hdl.handle.net/11234/1-2837>

⁶http://universaldependencies.org/conll18/conll18_ud_eval.py

Content	nsubj, obj, iobj, csubj, ccomp, xcomp, obl, vocative, expl, dislocated, advcl, advmod, discourse, nmod, appos, nummod, acl, amod, conj, fixed, flat, compound, list, parataxis, orphan, goeswith, reparandum, root, dep
Function	aux, cop, mark, det, clf, case, cc
Ignored	punct

Table 3: Universal dependency relations considered as pertaining to content words and function words, respectively, in MLAS. Content word relations are evaluated directly; words attached via functional relations are treated as features of their parent nodes.

Features	PronType, NumType, Poss, Reflex, Foreign, Abbr, Gender, Animacy, Number, Case, Definite, Degree, VerbForm, Mood, Tense, Aspect, Voice, Evident, Polarity, Person, Polite
----------	--

Table 4: Universal features whose values are evaluated in MLAS. Any other features are ignored.

universal dependency labels were taken into account, which means that language-specific subtypes such as `expl:pv` (pronoun of a pronominal verb), a subtype of the universal relation `expl` (expletive), were truncated to `expl` both in the gold standard and in the system output before comparing them.

In the end-to-end evaluation of our task, LAS is re-defined as the harmonic mean (F_1) of precision P and recall R , where

$$P = \frac{\#correctRelations}{\#systemNodes} \quad (1)$$

$$R = \frac{\#correctRelations}{\#goldNodes} \quad (2)$$

$$LAS = \frac{2PR}{P + R} \quad (3)$$

Note that attachment of all nodes including punctuation is evaluated. LAS is computed separately for each of the 82 test files and a macro-average of all these scores is used to rank the systems.

3.2 MLAS: Morphology-Aware Labeled Attachment Score

MLAS aims at cross-linguistic comparability of the scores. It is an extension of CLAS (Nivre and Fang, 2017), which was tested experimentally in the 2017 task. CLAS focuses on dependencies between content words and disregards attachment of function words; in MLAS, function words are not ignored, but they are treated as features of content words. In addition, part-of-speech tags and morphological features are evaluated, too.

The idea behind MLAS is that function words often correspond to morphological features in other languages. Furthermore, languages with many function words (e.g., English) have longer sentences than morphologically rich languages (e.g., Finnish), hence a single error in Finnish costs the parser significantly more than an error in English according to LAS.

The core part is identical to LAS (Section 3.1): for aligned system and gold nodes, their respective parent nodes are considered; if the system parent is not aligned with the gold parent, or if the universal relation label differs, the word is not counted as correctly attached. Unlike LAS, certain types of relations (Table 3) are not evaluated directly. Words attached via such relations (in either system or gold data) are not counted as independent words. Instead, they are treated as features of the content words they belong to. Therefore, a system-produced word counts as correct if it is aligned and attached correctly, its universal POS tag and selected morphological features (Table 4) are correct, all its function words are attached correctly, and their POS tags and features are also correct. Punctuation nodes are neither content nor function words; their attachment is ignored in MLAS.

3.3 BLEX: Bilexical Dependency Score

BLEX is similar to MLAS in that it focuses on relations between content words. Instead of morphological features, it incorporates lemmatization in the evaluation. It is thus closer to semantic content and evaluates two aspects of UD annota-

tion that are important for language understanding: dependencies and lexemes. The inclusion of this metric should motivate the competing teams to model lemmas, the last important piece of annotation that is not captured by the other metrics. A system that scores high in all three metrics will thus be a general-purpose language-analysis tool that tackles segmentation, morphology and surface syntax.

Computation of BLEX is analogous to LAS and MLAS. Precision and recall of correct attachments is calculated, attachment of function words and punctuation is ignored (Table 3). An attachment is correct if the parent and child nodes are aligned to the corresponding nodes in gold standard, if the universal dependency label is correct, and if the lemma of the child node is correct.

A few UD treebanks lack lemmatization (or, as in Uyghur, have lemmas only for some words and not for others). A system may still be able to predict the lemmas if it learns them in other treebanks. Such system should not be penalized just because no gold standard is available; therefore, if the gold lemma is a single underscore character (“_”), any system-produced lemma is considered correct.

3.4 Token Alignment

UD defines two levels of token/word segmentation. The lower level corresponds to what is usually understood as tokenization. However, unlike some popular tokenization schemes, it does not include any normalization of the non-whitespace characters. We can safely assume that any two tokenizations of a text differ only in whitespace while the remaining characters are identical. There is thus a 1-1 mapping between gold and system non-whitespace characters, and two tokens are aligned if all their characters match.

3.5 Syntactic Word Alignment

The higher segmentation level is based on the notion of *syntactic word*. Some languages contain *multi-word tokens* (MWT) that are regarded as contractions of multiple syntactic words. For example, the German token *zum* is a contraction of the preposition *zu* “to” and the article *dem* “the”.

Syntactic words constitute independent nodes in dependency trees. As shown by the example, it is not required that the MWT is a pure concatenation of the participating words; the simple token alignment thus does not work when MWTs

are involved. Fortunately, the CoNLL-U file format used in UD clearly marks all MWTs so we can detect them both in system output and in gold data. Whenever one or more MWTs have overlapping spans of surface character offsets, the longest common subsequence algorithm is used to align syntactic words within these spans.

3.6 Sentence Segmentation

Words are aligned and dependencies are evaluated in the entire file without considering sentence segmentation. Still, the accuracy of sentence boundaries has an indirect impact on attachment scores: any missing or extra sentence boundary necessarily makes one or more dependency relations incorrect.

3.7 Invalid Output

If a system fails to produce one of the 82 files or if the file is not valid CoNLL-U format, the score of that file (counting towards the system’s macro-average) is zero.

Formal validity is defined more leniently than for UD-released treebanks. For example, a non-existent dependency type does not render the whole file invalid, it only costs the system one incorrect relation. However, cycles and multi-root sentences are disallowed. A file is also invalid if there are character mismatches that could make the token-alignment algorithm fail.

3.8 Extrinsic Parser Evaluation

The metrics described above are all *intrinsic* measures: they evaluate the grammatical analysis task per se, with the hope that better scores correspond to output that is more useful for downstream NLP applications. Nevertheless, such correlations are not automatically granted. We thus seek to complement our task with an *extrinsic* evaluation, where the output of parsing systems is exploited by applications like biological event extraction, opinion analysis and negation scope resolution.

This optional track involves English only. It is organized in collaboration with the EPE initiative;⁷ for details see Fares et al. (2018).

4 TIRA: The System Submission Platform

Similarly to our 2017 task and to some other recent CoNLL shared tasks, we employed the cloud-

⁷<http://epe.nlpl.eu/>

based evaluation platform TIRA (Potthast et al., 2014),⁸ which implements the *evaluation as a service* paradigm (Hanbury et al., 2015). Instead of processing test data on their own hardware and submitting the outputs, participants submit working software. Naturally, software submissions bring about additional overhead for both organizers and participants, whereas the goal of an evaluation platform like TIRA is to reduce this overhead to a bearable level.

4.1 Blind Evaluation

Traditionally, evaluations in shared tasks are half-blind (the test data are shared with participants while the ground truth is withheld). TIRA enables fully blind evaluation, where the software is locked in a datalock together with the test data, its output is recorded but all communication channels to the outside are closed or tightly moderated. The participants do not even see the input to their software. This feature of TIRA was not too important in the present task, as UD data is not secret, and the participants were simply trusted that they would not exploit any knowledge of the test data they might have access to.

However, closing down all communication channels also has its downsides, since participants cannot check their running software; before the system run completes, even the task moderator does not see whether the system is really producing output and not just sitting in an endless loop. In order to alleviate this extra burden, we made two modifications compared to the previous year: 1. Participants were explicitly advised to invoke shorter runs that process only a subset of the test files. The organizers would then stitch the partial runs into one set of results. 2. Participants were able to see their scores on the test set rounded to the nearest multiple of 5%. This way they could spot anomalies possibly caused by ill-selected models. The exact scores remained hidden because we did not want the participants to fine-tune their systems against the test data.

4.2 Replicability

It is desirable that published experiments can be re-run yielding the same results, and that the algorithms can be tested on alternative test data in the future. Ensuring both requires that a to-be-evaluated software is preserved in working con-

dition for as long as possible. TIRA supplies participants with a virtual machine, offering a range of commonly used operating systems. Once deployed and tested, the virtual machines are archived to preserve the software within.

In addition, some participants agreed to share their code so that we decided to collect the respective projects in an open source repository hosted on GitHub.⁹

5 Baseline System

We prepared a set of baseline models using UDPipe 1.2 (Straka and Straková, 2017).

The baseline models were released together with the UD 2.2 training data. For each of the 73 treebanks with non-empty training data we trained one UDPipe model, utilizing training data for training and development data for hyperparameter tuning. If a treebank had no development data, we cut 10% of the training sentences and considered it as development data for the purpose of tuning hyperparameters of the baseline model (employing only the remainder of the original training data for the actual training in that case).

In addition to the treebank-specific models, we also trained a “mixed model” on samples from all treebanks. Specifically, we utilized the first 200 training sentences of each treebank (or less in case of small treebanks) as training data, and at most 20 sentences from each treebank’s development set as development data.

The baseline models, together with all information needed to replicate them (hyperparameters, the modified train-dev split where applicable, and pre-computed word embeddings for the parser) are available from <http://hdl.handle.net/11234/1-2859>.

Additionally, the released archive also contains the training and development data with predicted morphology. Morphology in development data was predicted using the baseline models, morphology in training data via “jack-knifing” (split the training set into 10 parts, train a model on 9 parts, use it to predict morphology in the tenth part, repeat for all 10 target parts). The same hyperparameters were used as those used to train the baseline model on the entire training set.

The UDPipe baseline models are able to reconstruct nearly all annotation from CoNLL-U files – they can generate segmentation, tokenization,

⁸<http://www.tira.io/>

⁹<https://github.com/CoNLL-UD-2018>

Treebank without training data	Substitution model
Breton KEB	mixed model
Czech PUD	Czech PDT
English PUD	English EWT
Faroese OFT	mixed model
Finnish PUD	Finnish TDT
Japanese Modern	Japanese GSD
Naija NSC	mixed model
Swedish PUD	Swedish Talbanken
Thai PUD	mixed model

Table 5: Substitution models of the baseline systems for treebanks without training data.

multi-word token splitting, morphological annotation (lemmas, UPOS, XPOS and FEATS) and dependency trees. Participants were free to use any part of the model in their systems – for all test sets, we provided UDPipe processed variants in addition to raw text inputs.

Baseline UDPipe Shared Task System The shared task baseline system employs the UDPipe 1.2 baseline models. For the nine treebanks without their own training data, a substitution model according to Table 5 was used.

6 Results

6.1 Official Parsing Results

Table 6 gives the main ranking of participating systems by the LAS F_1 score macro-averaged over all 82 test files. The table also shows the performance of the baseline UDPipe system; 17 of the 25 systems managed to outperform it. The baseline is comparatively weaker than in the 2017 task (only 12 out of 32 systems beat the baseline there). The ranking of the baseline system by MLAS is similar (Table 7) but in BLEX, the baseline jumps to rank 13 (Table 8). Besides the simple explanation that UDPipe 1.2 is good at lemmatization, we could also hypothesize that some teams put less effort in building lemmatization models (see also the last column in Table 10).

Each ranking has a different winning system, although the other two winners are typically closely following. The same 8–10 systems occupy best positions in all three tables, though with variable mutual ranking. Some teams seem to have deliberately neglected some of the evaluated attributes: Uppsala is rank 7 in LAS and MLAS, but 24 in

Team	LAS
1. HIT-SCIR (Che et al.)	75.84
2. TurkuNLP (Kanerva et al.)	73.28
3. UDPipe Future (Straka)	73.11
LATTICE (Lim et al.)	73.02
ICS PAS (Rybak and Wróblewska)	73.02
6. CEA LIST (Duthoo and Mesnard)	72.56
7. Uppsala (Smith et al.)	72.37
Stanford (Qi et al.)	72.29
9. AntNLP (Ji et al.)	70.90
NLP-Cube (Boroş et al.)	70.82
11. ParisNLP (Jawahar et al.)	70.64
12. SLT-Interactions (Bhat et al.)	69.98
13. IBM NY (Wan et al.)	69.11
14. UniMelb (Nguyen and Verspoor)	68.66
15. LeisureX (Li et al.)	68.31
16. KParse (Kirnap et al.)	66.58
17. Fudan (Chen et al.)	66.34
18. BASELINE UDPipe 1.2	65.80
19. Phoenix (Wu et al.)	65.61
20. CUNI x-ling (Rosa and Mareček)	64.87
21. BOUN (Özateş et al.)	63.54
22. ONLP lab (Seker et al.)	58.35
23. iParse (no paper)	55.83
24. HUJI (Hershovich et al.)	53.69
25. ArmParser (Arakelyan et al.)	47.02
26. SParse (Önder et al.)	1.95

Table 6: Ranking of the participating systems by the labeled attachment F_1 -score (LAS), macro-averaged over 82 test sets. Pairs of systems with significantly ($p < 0.05$) different LAS are separated by a line. Citations refer to the corresponding system-description papers in this volume.

BLEX; IBM NY is rank 13 in LAS but 24 in MLAS and 23 in BLEX.

While the LAS scores on individual treebanks are comparable to the 2017 task, the macro average is not, because the set of treebanks is different, and the impact of low-resource languages seems to be higher in the present task.

We used bootstrap resampling to compute 95% confidence intervals: they are in the range ± 0.11 to ± 0.16 (% LAS/MLAS/BLEX) for all systems except SParse (where it is ± 0.00).

Team	MLAS
1. UDPipe Future (Praha)	61.25
2. TurkuNLP (Turku)	60.99
Stanford (Stanford)	60.92
4. ICS PAS (Warszawa)	60.25
5. CEA LIST (Paris)	59.92
6. HIT-SCIR (Harbin)	59.78
7. Uppsala (Uppsala)	59.20
8. NLP-Cube (Bucureşti)	57.32
9. LATTICE (Paris)	57.01
10. AntNLP (Shanghai)	55.92
11. ParisNLP (Paris)	55.74
12. SLT-Interactions (Bengaluru)	54.52
13. LeisureX (Shanghai)	53.70
UniMelb (Melbourne)	53.62
15. KParse (İstanbul)	53.25
16. Fudan (Shanghai)	52.69
17. BASELINE UDPipe 1.2	52.42
Phoenix (Shanghai)	52.26
19. BOUN (İstanbul)	50.40
CUNI x-ling (Praha)	50.35
21. ONLP lab (Ra'anana)	46.09
22. iParse (Pittsburgh)	45.65
23. HUJI (Yerushalayim)	44.60
24. IBM NY (Yorktown Heights)	40.61
25. ArmParser (Yerevan)	36.28
26. SParse (İstanbul)	1.68

Table 7: Ranking of the participating systems by **MLAS**, macro-averaged over 82 test sets. Pairs of systems with significantly ($p < 0.05$) different MLAS are separated by a line.

We used paired bootstrap resampling to compute whether the difference between two neighboring systems is significant ($p < 0.05$).¹⁰

6.2 Secondary Metrics

In addition to the main LAS ranking, we evaluated the systems along multiple other axes, which may shed more light on their strengths and weaknesses. This section provides an overview of selected secondary metrics for systems matching or surpassing the baseline; a large number of additional results are available at the shared task website.¹¹

The website also features a LAS ranking of unofficial system runs, i.e. those that were not

¹⁰Using Udapi (Popel et al., 2017) eval.Conll18, marked by the presence or absence of horizontal lines in Tables 6–8.

¹¹<http://universaldependencies.org/conll18/results.html>

Team	BLEX
1. TurkuNLP (Turku)	66.09
2. HIT-SCIR (Harbin)	65.33
3. UDPipe Future (Praha)	64.49
ICS PAS (Warszawa)	64.44
5. Stanford (Stanford)	64.04
6. LATTICE (Paris)	62.39
CEA LIST (Paris)	62.23
8. AntNLP (Shanghai)	60.91
9. ParisNLP (Paris)	60.70
10. SLT-Interactions (Bengaluru)	59.68
11. UniMelb (Melbourne)	58.67
12. LeisureX (Shanghai)	58.42
13. BASELINE UDPipe 1.2	55.80
Phoenix (Shanghai)	55.71
15. NLP-Cube (Bucureşti)	55.52
16. KParse (İstanbul)	55.26
17. CUNI x-ling (Praha)	54.07
Fudan (Shanghai)	54.03
19. BOUN (İstanbul)	53.45
20. iParse (Pittsburgh)	48.71
21. HUJI (Yerushalayim)	48.05
22. ArmParser (Yerevan)	39.18
23. IBM NY (Yorktown Heights)	32.55
24. Uppsala (Uppsala)	32.09
25. ONLP lab (Ra'anana)	28.29
26. SParse (İstanbul)	1.71

Table 8: Ranking of the participating systems by **BLEX**, macro-averaged over 82 test sets. Pairs of systems with significantly ($p < 0.05$) different BLEX are separated by a line.

marked by their teams as primary runs, or were even run after the official evaluation phase closed and test data were unblinded. The difference from the official results is much less dramatic than in 2017, with the exception of the team SParse, who managed to fix their software and produce more valid output files.

As an experiment, we also applied the 2017 system submissions to the 2018 test data. This allows us to test how many systems can actually be used to produce new data without a glitch, as well as to see to what extent the results change over one year and two releases of UD. Here it should be noted that not all of the 2018 task languages and treebanks were present in the 2017 task, therefore causing many systems fail due to an unknown language or treebank code. The full results of this

Team	Toks	Wrds	Sents
1. Uppsala	97.60	98.18	83.80
2. HIT-SCIR	98.42	98.12	83.87
3. CEA LIST	98.16	97.78	82.79
4. CUNI x-ling	98.09	97.74	82.80
5. TurkuNLP	97.83	97.42	83.03
6. SLT-Interactions	97.51	97.09	83.01
7. UDPipe Future	97.46	97.04	83.64
8. Phoenix	97.46	97.03	82.91
9. BASELINE UDPipe	97.39	96.97	83.01
ParisNLP	97.39	96.97	83.01
AntNLP	97.39	96.97	83.01
UniMelb	97.39	96.97	83.01
BOUN	97.39	96.97	83.01
ICS PAS	97.39	96.97	83.01
LATTICE	97.39	96.97	83.01
LeisureX	97.39	96.97	83.01
KParse	97.39	96.97	83.01
18. Fudan	97.38	96.96	82.85
19. IBM NY	97.30	96.92	83.51
20. ONLP lab	97.28	96.86	83.00
21. NLP-Cube	97.36	96.80	82.55
22. Stanford	96.19	95.99	76.55
23. HUJI	94.95	94.61	80.84
24. ArmParser	79.75	79.41	13.33
25. iParse	78.45	78.11	68.37
26. SParse	2.32	2.32	2.34

Table 9: Tokenization, word segmentation and sentence segmentation (ordered by word F_1 scores; out-of-order scores in the other two columns are bold).

experiment are available on the shared task website.¹²

Table 9 evaluates detection of tokens, syntactic words and sentences. About a third of the systems trusted the baseline segmentation; this is less than in 2017. For most languages and in aggregate, the segmentation scores are very high and their impact on parsing scores is not easy to prove; but it likely played a role in languages where segmentation is hard. For example, HIT-SCIR’s word segmentation in Vietnamese surpasses the second system by a margin of 6 percent points; likewise, the system’s advantage in LAS and MLAS (but not in BLEX!) amounts to 7–8 points. Similarly, Uppsala and ParisNLP achieved good segmenta-

¹²<http://universaldependencies.org/conll18/results-2017-systems.html>

Team	UPOS	Feats	Lemm
1. Uppsala	90.91	87.59	58.50
2. HIT-SCIR	90.19	84.24	88.82
3. CEA LIST	89.97	86.83	88.90
4. TurkuNLP	89.81	86.70	91.24
5. LATTICE	89.53	83.74	87.84
6. UDPipe Future	89.37	86.67	89.32
7. Stanford	89.01	85.47	88.32
8. ICS PAS	88.70	85.14	87.99
9. CUNI x-ling	88.68	84.56	88.96
10. NLP-Cube	88.50	85.08	81.21
11. SLT-Interactions	88.12	83.72	87.51
12. IBM NY	88.02	59.11	59.51
13. UniMelb	87.90	83.74	87.84
14. KParse	87.62	84.32	86.26
15. Phoenix	87.49	83.87	87.69
16. ParisNLP	87.35	83.74	87.84
17. BASELINE UDPipe	87.32	83.74	87.84
AntNLP	87.32	83.74	87.84
19. ONLP lab	87.25	83.67	57.10
20. Fudan	87.25	83.47	85.91
21. BOUN	87.19	83.73	87.68
22. LeisureX	87.15	83.46	87.77
23. HUJI	85.06	81.51	85.61
24. ArmParser	72.99	69.91	72.22
25. iParse	71.38	68.64	71.68
26. SParse	2.25	2.29	2.28

Table 10: Universal POS tags, features and lemmas (ordered by UPOS F_1 scores; out-of-order scores in the other two columns are bold).

tion scores (better than their respective macro-averages) on Arabic. They were able to translate it into better LAS, but not MLAS and BLEX, where there were too many other chances to make an error.

The complexity of the new metrics, especially MLAS, is further underlined by Table 10: Uppsala is the clear winner in both UPOS tags and morphological features, but 6 other teams had better dependency relations and better MLAS. Note that as with segmentation, morphology predicted by the baseline system was available, though only a few systems seem to have used it without attempting to improve it.

6.3 Partial Results

Table 11 gives the three main scores averaged over the 61 “big” treebanks (training data larger than

Team	LAS	MLAS	BLEX
1. HIT-SCIR	84.37	70.12	75.05
2. Stanford	83.03	72.67	75.46
3. TurkuNLP	81.85	71.27	75.83
4. UDPipe Future	81.83	71.71	74.67
5. ICS PAS	81.72	70.30	74.42
6. CEA LIST	81.66	70.89	72.32
7. LATTICE	80.97	66.27	71.50
8. NLP-Cube	80.48	67.79	64.76
9. ParisNLP	80.29	65.88	70.95
10. Uppsala	80.25	68.81	36.02
11. SLT-Interactions	79.67	64.95	69.77
12. AntNLP	79.61	65.43	70.34
13. LeisureX	77.98	63.79	68.55
14. UniMelb	77.69	63.17	68.25
15. IBM NY	77.55	47.34	36.68
16. Fudan	75.42	62.28	62.90
17. KParse	74.84	62.40	63.84
18. BASELINE UDPipe	74.14	61.27	64.67
19. Phoenix	73.93	61.12	64.47
20. BOUN	72.85	60.00	62.99
21. CUNI x-ling	71.54	58.33	61.63
22. ONLP lab	67.08	55.20	33.08
23. iParse	66.55	55.37	58.80
24. HUJI	62.07	53.20	56.90
25. ArmParser	58.14	45.87	49.25
26. SParse	2.63	2.26	2.30

Table 11: Average LAS on the 61 “big” treebanks (ordered by LAS F_1 scores; out-of-order scores in the other two columns are bold).

test data, development data available). Higher scores reflect the fact that models for these test sets are easier to learn: enough data is available, no cross-lingual or cross-domain learning is necessary (the extra test sets are not included here). Regarding ranking, the Stanford system makes a remarkable jump when it does not have to carry the load of underresourced languages: from rank 8 to 2 in LAS, from 3 to 1 in MLAS and from 5 to 2 in BLEX.

Table 12 gives the LAS F_1 score on the nine low-resource languages only. Here we have a true specialist: The team CUNI x-ling lives up to its name and wins in all three scores, although in the overall ranking they fall even slightly behind the baseline. On the other hand, the scores are extremely low and the outputs are hardly useful for any downstream application. Especially morphol-

Team	LAS	MLAS	BLEX
1. CUNI x-ling	27.89	6.13	13.98
2. Uppsala	25.87	5.16	9.03
3. CEA LIST	23.90	3.75	10.99
4. HIT-SCIR	23.88	2.88	10.50
5. LATTICE	23.39	4.38	10.01
6. TurkuNLP	22.91	3.59	11.40
7. IBM NY	21.88	2.62	7.17
8. UDPipe Future	21.75	2.82	8.80
9. ICS PAS	19.26	1.89	6.17
10. AntNLP	18.59	3.43	8.61
11. KParse	17.84	3.32	6.58
12. SLT-Interactions	17.47	1.79	6.95
13. Stanford	17.45	2.76	7.63
14. BASELINE UDPipe	17.17	3.44	7.63
UniMelb	17.17	3.44	7.63
16. LeisureX	17.16	3.43	7.63
17. Phoenix	16.99	3.02	8.00
18. NLP-Cube	16.85	3.39	7.05
19. ParisNLP	16.52	2.53	6.75
20. ONLP lab	15.98	3.58	4.96
21. Fudan	15.45	2.98	6.61
22. BOUN	14.78	2.59	6.43
23. HUJI	8.53	0.92	2.77
24. ArmParser	7.47	1.86	3.54
25. iParse	2.82	0.23	0.97
26. SParse	0.00	0.00	0.00

Table 12: Average LAS, MLAS and BLEX on the 9 low-resource languages: Armenian (hy), Breton (br), Buryat (bxr), Faroese (fo), Kazakh (kk), Kurmanji (kmr), Naija (pcm), Thai (th) and Upper Sorbian (hsb) (ordered by LAS F_1 scores; out-of-order scores in the other two columns are bold).

ogy is almost impossible to learn from foreign languages, hence the much lower values of MLAS and BLEX. BLEX is a bit better than MLAS, which could be explained by cases where a word form is identical to its lemma. However, there are significant language-by-language differences; the best LAS on Faroese and Upper Sorbian surpassing 45%. This probably owes to the presence of many Germanic and Slavic treebanks in training data, including some of the largest datasets in UD. Three languages, Buryat, Kurmanji and Upper Sorbian, were introduced in the 2017 task as

Team	LAS	MLAS	BLEX
1. HIT-SCIR	69.53	45.94	53.30
2. LATTICE	68.12	45.03	51.71
3. ICS PAS	66.90	49.24	54.89
4. TurkuNLP	64.48	47.63	53.54
5. UDPipe Future	64.21	47.53	49.53
6. AntNLP	63.73	42.24	48.31
7. Uppsala	63.60	46.00	29.25
8. ParisNLP	60.84	40.71	46.08
9. CEA LIST	57.34	39.97	43.39
10. KParse	57.32	39.20	43.61
11. NLP-Cube	56.78	37.13	38.30
12. SLT-Interactions	56.74	35.73	42.90
13. IBM NY	56.13	26.51	25.23
14. UniMelb	56.12	36.09	42.09
15. BASELINE UDPipe	55.01	38.80	41.06
LeisureX	55.01	38.80	41.06
17. Phoenix	54.63	38.38	40.72
Fudan	54.63	38.15	40.07
19. CUNI x-ling	54.33	38.10	40.70
20. BOUN	50.18	34.29	36.75
21. Stanford	48.56	34.86	38.55
22. ONLP lab	47.49	32.74	22.39
23. iParse	38.79	28.03	29.62
24. HUJI	36.74	24.47	27.70
25. ArmParser	34.54	22.94	25.26
26. SParse	0.00	0.00	0.00

Table 13: Average attachment score on the 7 small treebanks: Galician TreeGal, Irish, Latin Perseus, North Sámi, Norwegian Nynorsk LIA, Russian Taiga and Slovenian SST (ordered by LAS F₁ scores; out-of-order scores in the other two columns are bold).

surprise languages and had higher scores there.¹³ This is because in 2017, the segmentation, POS tags and morphology UDPipe models were trained on the test data, applied to it via cross-validation, and made available to the systems. Such an approach makes the conditions unrealistic, therefore it was not repeated this year. Consequently, parsing these languages is now much harder.

In contrast, the results on the 7 treebanks with “small” training data and no development data (Table 13) are higher on average, but again the variance is significant. The smallest treebank

¹³The fourth surprise language, North Sámi, has now additional training data and does not fall in the low-resource category.

Team	LAS	MLAS	BLEX
1. HIT-SCIR	74.20	55.52	62.34
2. Stanford	73.14	58.75	61.96
3. LATTICE	72.34	55.60	60.42
4. Uppsala	72.27	57.80	29.73
5. ICS PAS	72.18	58.07	60.97
6. TurkuNLP	71.78	57.54	63.25
7. UDPipe Future	71.57	57.93	61.52
8. CEA LIST	70.45	54.99	57.83
9. NLP-Cube	69.83	55.01	54.15
10. IBM NY	69.40	46.59	38.12
11. AntNLP	68.87	53.47	57.71
12. UniMelb	68.72	52.05	56.77
13. Phoenix	66.97	52.26	55.69
14. BASELINE UDPipe	66.63	51.75	54.87
15. KParse	66.55	51.29	54.45
16. SLT-Interactions	64.73	48.47	54.90
17. CUNI x-ling	64.70	49.71	52.72
18. ParisNLP	64.09	48.79	53.16
19. Fudan	63.54	45.54	50.73
20. LeisureX	61.05	41.95	50.60
21. BOUN	56.46	41.91	45.12
22. HUJI	56.35	46.52	50.10
23. iParse	44.20	33.43	38.18
24. ONLP lab	43.33	30.20	20.08
25. ArmParser	0.00	0.00	0.00
SParse	0.00	0.00	0.00

Table 14: Average attachment score on the 5 additional test sets for high-resource languages: Czech PUD, English PUD, Finnish PUD, Japanese Modern and Swedish PUD (ordered by LAS F₁ scores; out-of-order scores in the other two columns are bold).

in the group, Norwegian Nynorsk LIA, has only 3583 training words. There are two larger Norwegian treebanks that could be used as additional training sources. However, the LIA treebank consists of spoken dialects and is probably quite dissimilar to the other treebanks. The same can be said about Slovenian SST and the other Slovenian treebank; SST is the most difficult dataset in the group, despite of having almost 20K of its own training words. Other treebanks, like Russian Taiga and Galician TreeGal, have much better scores (74% LAS, about 61% MLAS and 64% BLEX). There are also two treebanks that are the sole representatives of their languages: Irish and North Sámi. Their best LAS is around 70%: com-

parable to Nynorsk LIA but much better than SST. ICS PAS is the most successful system in the domain of small treebanks, especially when judged by MLAS and BLEX.

Table 14 gives the average LAS on the 5 extra test sets (no own training data, but other treebanks of the same language exist). Four of them come from the Parallel UD (PUD) collection introduced in the 2017 task (Zeman et al., 2017). The fifth, Japanese Modern, turned out to be one of the toughest test sets in this shared task. There is another Japanese treebank, GSD, with over 160K training tokens, but the Modern dataset seems almost inapproachable with models trained on GSD. A closer inspection reveals why: despite its name, it is actually a corpus of historical Japanese, although from the relatively recent Meiji and Taishō periods (1868–1926). An average sentence in GSD is about $1.3\times$ longer than in Modern. GSD has significantly more tokens tagged as auxiliaries, but more importantly, the top ten AUX lemmas in the two treebanks are completely disjoint sets. Some other words are out-of-vocabulary because their preferred spelling changed. For instance, the demonstrative pronoun *sore* is written using hiragana in GSD, but a kanji character is used in Modern. Striking differences can be observed also in dependency relations: in GSD, 3.7% relations are *nsubj* (subject), and 1.2% are *cop* (copula). In Modern, there is just 0.13% of subjects, and not a single occurrence of a copula.

See Tables 15, 16 and 17 for a ranking of all test sets by the best scores achieved on them by any parser. Note that this cannot be directly interpreted as a ranking of languages by their parsing difficulty: many treebanks have high ranks simply because the corresponding training data is large. Table 18 compares average LAS and MLAS for each treebank.

Finally, Tables 19 and 20 show the treebanks where word and sentence segmentation was extremely difficult (judged by the average parser score). Not surprisingly, word segmentation is difficult for the low-resource languages and for languages like Chinese, Vietnamese, Japanese and Thai, where spaces do not separate words. Notably the Japanese GSD set is not as difficult, but whoever trusted it, crashed on the “Modern” set. Sentence segmentation was particularly hard for treebanks without punctuation, i.e., most of the classical languages and spoken data.

Treebank	LAS	Best system	Avg	StdDev
1. pl_lfg	94.86	HIT-SCIR	85.89	± 6.97
2. ru_syntagrus	92.48	HIT-SCIR	79.68	± 9.09
3. hi_hdtb	92.41	HIT-SCIR	85.16	± 5.32
4. pl_sz	92.23	HIT-SCIR	81.47	± 7.27
5. cs_fictree	92.02	HIT-SCIR	82.10	± 7.26
6. it_isdt	92.00	HIT-SCIR	87.61	± 4.12
7. cs_pdt	91.68	HIT-SCIR	82.18	± 6.91
8. ca_ancora	91.61	HIT-SCIR	83.61	± 6.01
9. cs_cac	91.61	HIT-SCIR	82.69	± 6.93
10. sl_ssj	91.47	HIT-SCIR	75.00	± 9.13
11. no_bokmaal	91.23	HIT-SCIR	79.80	± 7.29
12. bg_btb	91.22	HIT-SCIR	82.52	± 5.88
13. no_nynorsk	90.99	HIT-SCIR	78.55	± 7.88
14. es_ancora	90.93	HIT-SCIR	82.84	± 6.17
15. fi_pud	90.23	HIT-SCIR	68.87	±15.61
16. fr_sequoia	89.89	LATTICE	80.55	± 5.91
17. el_gdt	89.65	HIT-SCIR	80.65	± 6.05
18. nl_alpino	89.56	HIT-SCIR	77.76	± 7.42
19. sk_snk	88.85	HIT-SCIR	76.53	± 7.24
20. fi_tdt	88.73	HIT-SCIR	73.55	± 9.39
21. sr_set	88.66	Stanford	79.84	± 6.57
22. sv_talbanken	88.63	HIT-SCIR	77.71	± 6.50
23. fi_ftb	88.53	HIT-SCIR	76.89	± 7.60
24. uk_iu	88.43	HIT-SCIR	72.47	± 8.25
25. fa_seraji	88.11	HIT-SCIR	78.71	± 6.04
26. en_pud	87.89	LATTICE	74.51	± 8.28
27. pt_bosque	87.81	Stanford	80.49	± 5.46
28. hr_set	87.36	HIT-SCIR	78.37	± 6.42
29. fro_sremf	87.12	UDPipe Future	74.38	±16.74
30. la_itb	87.08	HIT-SCIR	77.00	± 7.42
31. ko_kaist	86.91	HIT-SCIR	77.10	± 8.72
32. fr_gsd	86.89	HIT-SCIR	79.43	± 5.47
33. ro_rrt	86.87	HIT-SCIR	75.77	± 7.66
34. nl_lassysmall	86.84	HIT-SCIR	75.08	± 6.59
35. da_ddt	86.28	HIT-SCIR	75.02	± 6.47
36. cs_pud	86.13	HIT-SCIR	73.24	± 9.97
37. af_afribooms	85.47	HIT-SCIR	76.61	± 6.17
38. et_edt	85.35	HIT-SCIR	72.08	± 8.71
39. ko_gsd	85.14	HIT-SCIR	71.88	±10.53
40. en_gum	85.05	LATTICE	74.20	± 6.27
41. en_ewt	84.57	HIT-SCIR	75.99	± 5.40
42. eu_bdt	84.22	HIT-SCIR	72.08	± 8.83
43. sv_lines	84.08	HIT-SCIR	73.76	± 5.98
44. lv_lvtb	83.97	HIT-SCIR	67.76	± 9.01
45. ur_udtb	83.39	HIT-SCIR	75.89	± 4.69
46. ja_gsd	83.11	HIT-SCIR	73.68	± 4.55
47. gl_ctg	82.76	Stanford	72.46	± 7.13
48. hu_szeged	82.66	HIT-SCIR	67.05	± 8.63
49. en_lines	81.97	HIT-SCIR	72.28	± 5.59
50. de_gsd	80.36	HIT-SCIR	70.13	± 7.14
51. sv_pud	80.35	HIT-SCIR	67.02	± 9.23
52. id_gsd	80.05	HIT-SCIR	73.05	± 4.69
53. it_postwita	79.39	HIT-SCIR	64.95	± 6.88
54. grc_perseus	79.39	HIT-SCIR	59.01	±15.56
55. grc_proiel	79.25	HIT-SCIR	65.02	±14.58
56. ar_padt	77.06	Stanford	64.07	± 6.41
57. zh_gsd	76.77	HIT-SCIR	60.32	± 6.14
58. he_jtb	76.09	Stanford	58.73	± 5.29
59. fr_spoken	75.78	HIT-SCIR	64.66	± 5.35
60. cu_proiel	75.73	Stanford	62.64	± 6.98
61. gl_treegal	74.25	UDPipe Future	64.65	± 5.61
62. ru_taiqa	74.24	ICS PAS	56.27	± 9.16
63. la_proiel	73.61	HIT-SCIR	61.25	± 6.87
64. la_perseus	72.63	HIT-SCIR	46.91	±11.12
65. ga_idt	70.88	TurkuNLP	58.37	± 7.05
66. no_nynorskliia	70.34	HIT-SCIR	50.33	± 9.28
67. sme_giella	69.87	LATTICE	51.10	±14.32
68. got_proiel	69.55	Stanford	60.55	± 4.93
69. ug_udt	67.05	HIT-SCIR	54.27	± 6.90
70. tr_imst	66.44	HIT-SCIR	55.61	± 6.49
71. sl_sst	61.39	HIT-SCIR	47.07	± 5.84
72. vi_vtb	55.22	HIT-SCIR	40.40	± 4.43
73. fo_ofst	49.43	CUNI x-ling	27.87	± 9.75
74. hsb_ufal	46.42	SLT-Interactions	26.48	± 8.90
75. br_keb	38.64	CEA LIST	13.27	± 8.77
76. hy_armtdp	37.01	LATTICE	22.39	± 7.91
77. kk_ktb	31.93	Uppsala	19.11	± 6.34
78. kmr_mg	30.41	IBM NY	20.27	± 6.14
79. pcm_nsc	30.07	CUNI x-ling	13.19	± 5.76
80. ja_modern	28.33	Stanford	18.92	± 5.14
81. bxr_bdt	19.53	AntNLP	11.45	± 4.28
82. th_pud	13.70	CUNI x-ling	1.38	± 2.83

Table 15: Treebank ranking by best parser LAS (Avg=average LAS over all systems, out-of-order scores in bold).

Treebank	MLAS	Best system	Avg	StDev
1. pl_lfg	86.93	UDPipe Future	73.73	± 7.29
2. ru_syntagrus	86.76	UDPipe Future	71.63	± 9.36
3. cs_pdt	85.10	UDPipe Future	73.61	± 6.32
4. cs_fictree	84.23	ICS PAS	69.91	± 7.77
5. ca_ancora	84.07	UDPipe Future	74.62	± 7.69
6. es_ancora	83.93	Stanford	74.61	± 7.43
7. it_isdt	83.89	Stanford	77.14	± 8.89
8. fi_pud	83.78	Stanford	62.38	± 14.83
9. no_bokmaal	83.68	UDPipe Future	70.75	± 8.92
10. cs_cac	83.42	UDPipe Future	71.39	± 6.89
11. bg_btb	83.12	UDPipe Future	73.18	± 7.15
12. fr_sequoia	82.55	Stanford	70.42	± 9.04
13. sl_ssj	82.38	Stanford	62.41	± 9.18
14. no_nynorsk	81.86	UDPipe Future	68.62	± 9.45
15. ko_kaist	81.29	HIT-SCIR	70.18	± 9.36
16. ko_gsd	80.85	HIT-SCIR	63.73	± 16.02
17. fi_tdt	80.84	Stanford	65.27	± 9.22
18. fa_seraji	80.83	UDPipe Future	71.23	± 7.77
19. pl_sz	80.77	Stanford	64.80	± 8.49
20. fro_srcmf	80.28	UDPipe Future	65.19	± 16.58
21. la_itlb	79.84	ICS PAS	67.77	± 8.37
22. fi_ftb	79.65	TurkuNLP	66.11	± 8.86
23. sv_talbanken	79.32	Stanford	68.05	± 8.49
24. ro_rrt	78.68	TurkuNLP	67.43	± 7.24
25. el_gdt	78.66	Stanford	64.29	± 8.28
26. fr_gsd	78.44	Stanford	69.33	± 8.59
27. hi_hdtb	78.30	UDPipe Future	68.48	± 5.88
28. sr_set	77.73	UDPipe Future	67.33	± 5.96
29. da_ddt	77.31	Stanford	65.00	± 6.89
30. et_edt	76.97	TurkuNLP	63.59	± 8.34
31. nl_alpino	76.52	Stanford	62.82	± 9.81
32. en_ewt	76.33	Stanford	66.84	± 5.86
33. pt_bosque	75.94	Stanford	66.22	± 6.76
34. cs_pud	75.81	UDPipe Future	60.47	± 11.36
35. af_afribooms	75.67	UDPipe Future	63.76	± 7.06
36. sk_snk	75.01	Stanford	56.82	± 8.32
37. en_pud	74.86	Stanford	63.05	± 7.89
38. nl_lassysmall	74.11	Stanford	61.95	± 9.12
39. hr_set	73.44	Stanford	60.08	± 7.07
40. en_gum	73.24	ICS PAS	61.72	± 7.69
41. ja_gsd	72.62	HIT-SCIR	59.52	± 6.20
42. uk_iu	72.27	UDPipe Future	55.45	± 8.08
43. en_lines	72.25	ICS PAS	62.35	± 8.04
44. eu_bdt	71.73	UDPipe Future	58.49	± 8.62
45. gl_ctg	70.92	Stanford	57.92	± 14.10
46. ar_padt	68.54	Stanford	53.28	± 6.12
47. it_postwita	68.50	Stanford	51.72	± 8.80
48. id_gsd	68.36	Stanford	61.03	± 6.49
49. lv_lvtb	67.89	Stanford	53.31	± 7.96
50. hu_szeged	67.13	UDPipe Future	53.08	± 8.01
51. zh_gsd	66.62	HIT-SCIR	50.42	± 5.87
52. sv_lines	66.58	Stanford	57.40	± 7.43
53. fr_spoken	64.67	HIT-SCIR	53.17	± 5.61
54. he_htb	63.38	Stanford	45.22	± 4.94
55. cu_proiel	63.31	Stanford	50.28	± 6.69
56. ru_taiqa	61.59	ICS PAS	37.16	± 7.53
57. gl_treegal	60.63	UDPipe Future	47.35	± 5.93
58. grc_proiel	60.27	Stanford	47.62	± 11.82
59. la_proiel	59.36	Stanford	47.79	± 6.90
60. de_gsd	58.04	TurkuNLP	39.13	± 10.35
61. ur_udtb	57.98	TurkuNLP	49.64	± 4.21
62. no_nynorskliia	57.51	ICS PAS	37.08	± 7.78
63. sme_giella	57.47	TurkuNLP	38.29	± 12.37
64. got_proiel	56.45	UDPipe Future	46.18	± 5.36
65. tr_imst	55.73	Stanford	45.26	± 6.15
66. grc_perseus	54.98	HIT-SCIR	35.65	± 12.31
67. sv_pud	51.74	TurkuNLP	39.41	± 7.78
68. la_perseus	49.77	ICS PAS	28.67	± 8.06
69. vi_vtb	47.61	HIT-SCIR	32.45	± 7.28
70. sl_sst	45.93	ICS PAS	33.12	± 5.33
71. ga_idt	45.79	TurkuNLP	33.70	± 5.18
72. ug_udt	45.78	UDPipe Future	35.08	± 5.96
73. br_keb	13.91	Uppsala	1.52	± 3.34
74. hy_armtdp	13.36	CUNI x-ling	5.94	± 2.92
75. ja_modern	11.82	Uppsala	6.45	± 2.59
76. hsb_ufal	9.09	LATTICE	4.66	± 2.37
77. kk_ktb	8.93	CUNI x-ling	5.04	± 2.34
78. kmr_mg	7.98	IBM NY	4.01	± 1.96
79. th_pud	6.29	CUNI x-ling	0.42	± 1.27
80. pcm_nsc	5.30	KParse	3.00	± 1.30
81. bxr_bdt	2.98	AntNLP	1.33	± 0.72
82. fo_of	1.07	CUNI x-ling	0.37	± 0.21

Table 16: Treebank ranking by best parser MLAS.

Treebank	BLEX	Best system	Avg	StDev
1. pl_lfg	90.42	TurkuNLP	72.81	± 16.96
2. ru_syntagrus	88.65	TurkuNLP	68.57	± 18.07
3. cs_pdt	87.91	HIT-SCIR	74.41	± 14.88
4. cs_fictree	87.81	ICS PAS	71.10	± 16.26
5. cs_cac	86.79	TurkuNLP	71.61	± 18.18
6. hi_hdtb	86.74	HIT-SCIR	75.80	± 9.28
7. pl_sz	86.29	TurkuNLP	67.33	± 17.15
8. no_bokmaal	85.82	UDPipe Future	69.52	± 13.54
9. ca_ancora	85.47	UDPipe Future	72.60	± 12.31
10. es_ancora	84.92	HIT-SCIR	72.10	± 12.71
11. it_isdt	84.76	ICS PAS	75.42	± 10.72
12. fr_sequoia	84.67	ICS PAS	70.63	± 11.66
13. no_nynorsk	84.44	TurkuNLP	67.43	± 14.10
14. la_itlb	84.37	TurkuNLP	68.10	± 17.85
15. bg_btb	84.31	TurkuNLP	68.13	± 15.02
16. fro_srcmf	84.11	UDPipe Future	70.46	± 16.40
17. sr_set	83.28	TurkuNLP	65.62	± 17.61
18. sl_ssj	83.23	TurkuNLP	62.54	± 17.20
19. fi_ftb	82.44	TurkuNLP	59.66	± 16.50
20. fi_pud	82.44	TurkuNLP	52.25	± 18.50
21. sv_talbanken	81.44	TurkuNLP	66.45	± 13.18
22. fi_tdt	81.24	TurkuNLP	54.70	± 17.25
23. fr_gsd	81.18	HIT-SCIR	69.61	± 10.58
24. ro_rrt	80.97	TurkuNLP	63.53	± 15.84
25. sk_snk	80.74	TurkuNLP	58.35	± 15.07
26. pt_bosque	80.62	TurkuNLP	68.71	± 11.27
27. en_pud	80.53	LATTICE	64.73	± 10.88
28. cs_pud	80.53	ICS PAS	64.62	± 16.03
29. hr_set	80.50	TurkuNLP	64.64	± 17.13
30. fa_seraji	80.44	Stanford	68.38	± 7.39
31. el_gdt	80.09	TurkuNLP	63.26	± 15.60
32. ko_kaist	79.55	TurkuNLP	57.32	± 20.78
33. et_edt	79.37	TurkuNLP	57.06	± 16.14
34. nl_alpino	79.15	HIT-SCIR	64.29	± 10.83
35. en_ewt	78.44	HIT-SCIR	67.53	± 8.47
36. uk_iu	78.38	TurkuNLP	57.78	± 15.95
37. eu_bdt	78.15	TurkuNLP	60.52	± 15.24
38. da_ddt	78.07	TurkuNLP	63.16	± 11.41
39. sv_lines	77.01	ICS PAS	63.13	± 11.72
40. id_gsd	76.56	Stanford	62.52	± 7.89
41. nl_lassysmall	76.54	HIT-SCIR	60.92	± 11.93
42. af_afribooms	76.44	TurkuNLP	63.87	± 9.62
43. ko_gsd	76.31	TurkuNLP	54.13	± 17.78
44. en_lines	75.29	HIT-SCIR	62.29	± 9.27
45. gl_ctg	75.14	Stanford	60.86	± 10.82
46. ur_udtb	73.79	TurkuNLP	62.93	± 6.42
47. ja_gsd	73.79	HIT-SCIR	60.87	± 6.04
48. en_gum	73.57	ICS PAS	61.02	± 8.59
49. hu_szeged	73.17	TurkuNLP	55.42	± 10.95
50. zh_gsd	72.97	HIT-SCIR	55.66	± 6.26
51. lv_lvtb	72.40	TurkuNLP	53.42	± 14.56
52. de_gsd	71.40	HIT-SCIR	54.86	± 14.99
53. cu_proiel	71.31	Stanford	51.27	± 15.35
54. ar_padt	70.06	Stanford	49.13	± 18.98
55. it_postwita	69.34	HIT-SCIR	50.97	± 8.76
56. grc_proiel	69.03	TurkuNLP	48.58	± 19.91
57. la_proiel	67.60	TurkuNLP	51.03	± 14.56
58. sv_pud	66.12	TurkuNLP	50.20	± 11.30
59. fr_spoken	65.63	HIT-SCIR	52.57	± 7.29
60. he_htb	65.04	Stanford	47.22	± 6.60
61. ru_taiqa	64.36	ICS PAS	39.32	± 10.49
62. gl_treegal	64.29	UDPipe Future	49.38	± 8.18
63. got_proiel	63.98	Stanford	48.79	± 13.77
64. no_nynorskliia	60.98	ICS PAS	41.20	± 8.64
65. tr_imst	60.13	TurkuNLP	45.39	± 10.38
66. sme_giella	60.10	TurkuNLP	35.76	± 12.68
67. grc_perseus	58.68	TurkuNLP	36.48	± 16.03
68. ug_udt	55.42	HIT-SCIR	41.64	± 8.09
69. ga_idt	55.18	TurkuNLP	37.83	± 7.61
70. la_perseus	52.75	ICS PAS	30.16	± 11.05
71. sl_sst	50.94	ICS PAS	37.20	± 6.87
72. vi_vtb	44.02	Stanford	35.50	± 3.74
73. pcm_nsc	26.04	CUNI x-ling	12.07	± 5.63
74. hsb_ufal	21.09	LATTICE	11.26	± 4.97
75. br_keb	20.70	TurkuNLP	4.19	± 4.93
76. hy_armtdp	19.04	CUNI x-ling	10.68	± 4.37
77. fo_of	14.40	CUNI x-ling	7.32	± 3.33
78. ja_modern	13.79	Stanford	7.70	± 2.86
79. kmr_mg	13.66	LATTICE	8.44	± 3.11
80. kk_ktb	11.33	CUNI x-ling	6.75	± 2.95
81. th_pud	10.77	CUNI x-ling	0.91	± 2.11
82. bxr_bdt	6.65	AntNLP	3.39	± 1.61

Table 17: Treebank ranking by best parser BLEX.

Treebank	LAS	MLAS	Diff	Language
1. de_gsd	70.13	39.13	31.01	German
2. sv_pud	67.02	39.41	27.61	Swedish
3. fo_of	27.87	0.37	27.50	Faroese
4. ur_udtb	75.89	49.64	26.25	Urdu
5. ga_idt	58.37	33.70	24.66	Irish
6. grc_perseus	59.01	35.65	23.36	Ancient Greek
7. hsb_ufal	26.48	4.66	21.82	Upper Sorbian
8. sk_snk	76.53	56.82	19.71	Slovak
9. ug_udt	54.27	35.08	19.20	Uyghur
10. ru_taiga	56.27	37.16	19.12	Russian
11. hr_set	78.37	60.08	18.29	Croatian
12. la_perseus	46.91	28.67	18.24	Latin
13. grc_proiel	65.02	47.62	17.40	Ancient Greek
14. gl_treegal	64.65	47.35	17.30	Galician
15. uk_iu	72.47	55.45	17.01	Ukrainian
16. hi_hdtb	85.16	68.48	16.68	Hindi
17. pl_sz	81.47	64.80	16.67	Polish
18. hy_armtdp	22.39	5.94	16.45	Armenian
19. el_gdt	80.65	64.29	16.36	Greek
20. sv_lines	73.76	57.40	16.36	Swedish
21. kmr_mg	20.27	4.01	16.26	Kurmanji
22. nl_alpino	77.76	62.82	14.95	Dutch
23. gl_ctg	72.46	57.92	14.55	Galician
24. lv_lvtb	67.76	53.31	14.45	Latvian
25. got_proiel	60.55	46.18	14.37	Gothic
26. pt_bosque	80.49	66.22	14.27	Portuguese
27. ja_gsd	73.68	59.52	14.16	Japanese
28. kk_ktb	19.11	5.04	14.07	Kazakh
29. hu_szeged	67.05	53.08	13.96	Hungarian
30. sl_sst	47.07	33.12	13.95	Slovenian
31. eu_bdt	72.08	58.49	13.59	Basque
32. he_htb	58.73	45.22	13.51	Hebrew
33. la_proiel	61.25	47.79	13.46	Latin
34. no_nynorsk	50.33	37.08	13.25	Norwegian
35. it_postwita	64.95	51.72	13.22	Italian
36. nl_lassysmall	75.08	61.95	13.14	Dutch
37. af_afribooms	76.61	63.76	12.84	Afrikaans
38. sme_giella	51.10	38.29	12.82	North Sámi
39. cs_pud	73.24	60.47	12.77	Czech
40. sl_ssj	75.00	62.41	12.59	Slovenian
41. sr_set	79.84	67.33	12.50	Serbian
42. en_gum	74.20	61.72	12.48	English
43. ja_modern	18.92	6.45	12.47	Japanese
44. cu_proiel	62.64	50.28	12.36	Old Church Slavonic
45. cs_fictree	82.10	69.91	12.19	Czech
46. pl_lfg	85.89	73.73	12.17	Polish
47. id_gsd	73.05	61.03	12.02	Indonesian
48. br_keb	13.27	1.52	11.75	Breton
49. fr_spoken	64.66	53.17	11.49	French
50. en_pud	74.51	63.05	11.46	English
51. cs_cac	82.69	71.39	11.29	Czech
52. ar_padt	64.07	53.28	10.79	Arabic
53. fi_ftb	76.89	66.11	10.78	Finnish
54. it_isdt	87.61	77.14	10.47	Italian
55. tr_imst	55.61	45.26	10.34	Turkish
56. pcm_nsc	13.19	3.00	10.19	Naija
57. fr_sequoia	80.55	70.42	10.13	French
58. bxr_bdt	11.45	1.33	10.12	Buryat
59. fr_gsd	79.43	69.33	10.10	French
60. da_dtd	75.02	65.00	10.02	Danish
61. no_nynorsk	78.55	68.62	9.93	Norwegian
62. en_lines	72.28	62.35	9.93	English
63. zh_gsd	60.32	50.42	9.90	Chinese
64. sv_talbanken	77.71	68.05	9.66	Swedish
65. bg_btb	82.52	73.18	9.34	Bulgarian
66. la_itb	77.00	67.77	9.23	Latin
67. fro_srcmf	74.38	65.19	9.18	Old French
68. en_ewt	75.99	66.84	9.15	English
69. no_bokmaal	79.80	70.75	9.05	Norwegian
70. ca_ancora	83.61	74.62	8.99	Catalan
71. cs_pdt	82.18	73.61	8.57	Czech
72. et_edt	72.08	63.59	8.50	Estonian
73. ro_rrt	75.77	67.43	8.33	Romanian
74. fi_tdt	73.55	65.27	8.28	Finnish
75. es_ancora	82.84	74.61	8.23	Spanish
76. ko_gsd	71.88	63.73	8.15	Korean
77. ru_syntagrus	79.68	71.63	8.05	Russian
78. vi_vtb	40.40	32.45	7.95	Vietnamese
79. fa_seraji	78.71	71.23	7.48	Persian
80. ko_kaist	77.10	70.18	6.92	Korean
81. fi_pud	68.87	62.38	6.49	Finnish
82. th_pud	1.38	0.42	0.96	Thai

Table 18: Treebank ranking by difference between average parser LAS and MLAS.

Treebank	Best	Best system	Avg	StDev
70. bxr_bdt	99.24	IBM NY	88.64	± 8.09
71. fi_pud	99.69	Uppsala	88.13	±10.81
72. zh_gsd	96.71	HIT-SCIR	86.91	± 3.83
73. fo_of	99.47	CUNI x-ling	86.76	±10.68
74. ar_padt	96.81	Stanford	86.62	± 7.00
75. kmr_mg	96.97	Uppsala	86.61	± 7.16
76. kk_ktb	97.40	Uppsala	85.55	± 7.45
77. br_keb	92.45	TurkuNLP	83.76	± 7.37
78. he_htb	93.98	Stanford	82.45	± 3.80
79. vi_vtb	93.46	HIT-SCIR	81.71	± 3.73
80. pcm_nsc	99.71	CEA LIST	79.94	±10.69
81. ja_modern	75.69	HIT-SCIR	59.40	± 7.70
82. th_pud	69.93	Uppsala	17.16	±20.57

Table 19: Treebanks with most difficult word segmentation (by average parser F_1).

Treebank	Best	Best system	Avg	StDev
73. grc_proiel	51.84	HIT-SCIR	42.46	± 7.33
74. cu_proiel	48.67	Stanford	35.54	± 4.02
75. la_proiel	39.61	Stanford	33.40	± 5.39
76. got_proiel	38.23	Stanford	27.22	± 4.47
77. it_postwita	65.90	Stanford	25.25	±14.30
78. sl_sst	24.43	NLP-Cube	20.92	± 4.70
79. fr_spoken	24.17	Stanford	20.43	± 2.89
80. th_pud	12.37	TurkuNLP	1.75	± 3.68
81. pcm_nsc	0.93	Stanford	0.06	± 0.19
82. ja_modern	0.23	Stanford	0.01	± 0.04

Table 20: Treebanks with most difficult sentence segmentation (by average parser F_1).

7 Analysis of Submitted Systems

Table 21 gives an overview of 24 of the systems evaluated in the shared task. The overview is based on a post-evaluation questionnaire to which 24 of 25 teams responded. Systems are ordered alphabetically by name and their LAS rank is indicated in the second column.

Looking first at word and sentence segmentation, we see that, while a clear majority of systems (19/24) rely on the baseline system for segmentation, slightly more than half (13/24) have developed their own segmenter, or tuned the baseline segmenter, for at least a subset of languages. This is a development from 2017, where only 7 out of 29 systems used anything other than the baseline segmenter.

When it comes to morphological analysis, including universal POS tags, features and lemmas, all systems this year include some such component, and only 6 systems rely entirely on the base-

System	R	Segment	Morph	Syntax	WEmb	Additional Data	MultiLing
AntNLP	9	Base	Base	Single-G	FB	None	Own _S
ArmParser	25	Base	Own	Single	FB	None	None
BOUN	21	Base	Base	Single-T	Base	None	None
CEA LIST	6	Base	B _L /Own	Single-G/T	B/FB	OPUS/Wikt	Own _L
CUNI x-ling	20	B/Own	B/Own	Single/Ens	FB/None	O/UM/WALS/Wiki	Own _{L,S}
Fudan	17	Base	Base	Ensemble	None	None	Own _{L,S}
HIT-SCIR	1	B/Own	Base	Ensemble	B/FB/Crawl	None	Own _{L,S}
HUJI	24	Base	Base	Single-T	FB	None	Own _L
IBM NY	13	B/Own	B/Joint	Ensemble-T	B/FB	Wiki	Own _{L,S}
ICS PAS	3	Base	Own	Single-G	FB/None	None	None
KParse	16	B/Own	Own	Single	Other	None	Own _L
LATTICE	3	Base	Own _U	Single-G/Ens	B/FB/Crawl	OPUS/Wiki	Own _{L,S}
LeisureX	15	Base	Own	Single	Base	None	Own _L
NLP-Cube	9	Own	Own	Single	FB	None	Own _L
ONLP lab	22	Base	Base	Single-T	None	UML	None
ParisNLP	11	B/Own	B/Own	Single-G	FB	UML	Own _L
Phoenix	19	Own	Own _U	Single	Train	None	Own _L
SLT-Interactions	12	B/Own	Own	Single	Crawl	None	Own _L
SParse	26	B/Own	Own	Single-G	Crawl	None	Own _L
Stanford	7	Own	Own	Single-G	B/FB	None	None
TurkuNLP	2	B/Own	Own	Single-G	B/FB	OPUS/Aper	Own _L
UDPipe Future	3	Own	Joint	Single-G	B/FB	None	None
UniMelb	14	Base	Joint	Single	Base	None	Base
Uppsala	7	Own	Own _{U,F}	Single-T	B/FB/Wiki	OPUS/Wiki/Aper	Own _{L,S}

Table 21: Classification of participating systems. **R** = LAS ranking. **Segment** = word/sentence segmentation. **Morph** = morphological analysis, including universal POS tags [U], features [F] and lemmas [L], with subscripts for subsets [Joint = morphological component trained jointly with syntactic parser]. **Syntax** = syntactic parsing [Single = single parser; Ensemble (or Ens) = parser ensemble; G = graph-based; T = transition-based]. **WEmb** = pre-trained word embeddings [FB = Facebook; Crawl = trained on web crawl data provided by the organizers; Wiki = trained on Wikipedia data; Train = trained on treebank training data]. **Additional Data** = data used in addition to treebank training sets [OPUS (or O) = OPUS, Aper = Apertium morphological analysers, Wikt = Wiktionary, Wiki = Wikipedia, UM = UniMorph, UML = Universal Morphological Lattices, WALS = World Atlas of Language Structures]. **MultiLing** = multilingual models used for low-resource (L) or small (S) languages. In all columns, Base (or B) refers to the Baseline UDPipe system or the baseline word embeddings provided by the organizers, while None means that there is no corresponding component in the system.

line UDPipe system. This is again quite different from 2017, where more than half the systems either just relied on the baseline tagger (13 systems) or did not predict any morphology at all (3 systems). We take this to be primarily a reflection of the fact that two out of three official metrics included (some) morphological analysis this year, although 3 systems did not predict the lemmas required for the BLEXP metric (and 2 systems only predicted universal POS tags, no features). As far as we can tell from the questionnaire responses,

only 3 systems used a model where morphology and syntax were predicted jointly.¹⁴

For syntactic parsing, most teams (19) use a single parsing model, while 5 teams, including the winning HIT-SCIR system, build ensemble models, either for all languages or a subset of them. When it comes to the type of parsing model, we observe that graph-based models are more popular than transition-based models this year, while the opposite was true in 2017. We hypothesize that

¹⁴The ONLP lab system also has a joint model but in the end used the baseline morphology as it gave better results.

this is due to the superior performance of the Stanford graph-based parser in last year’s shared task, and many of the high-performing systems this year either incorporate that parser or a reimplementa-tion of it.¹⁵

The majority of parsers make use of pre-trained word embeddings. Most popular are the Facebook embeddings, which are used by 17 systems, fol-lowed by the baseline embeddings provided by the organizers (11), and embeddings trained on web crawl data (4).¹⁶ When it comes to additional data, over and above the treebank training sets and pre-trained word embeddings, the most striking obser-vation is that a majority of systems (16) did not use any at all. Those that did primarily used OPUS (5), Wikipedia dumps (3), Apertium morpholog-ical analyzers (2), and Universal Morphological Lattices (2). The CUNI x-ling system, which fo-cused on low-resource languages, also exploited UniMorph and WALS (in addition to OPUS and Wikipedia).

Finally, we note that a majority of systems make use of models trained on multiple languages to improve parsing for languages with little or no training data. According to the questionnaire re-sponses, 15 systems use multilingual models for the languages classified as “low-resource”, while 7 systems use them for the languages classified as “small”.¹⁷ Only one system relied on the baseline dellexicalized parser trained on data from all lan-guages.

8 Conclusion

The CoNLL 2018 Shared Task on UD parsing, the second in the series, was novel in several respects. Besides using cross-linguistically consistent lin-guistic representations, emphasizing end-to-end processing of text, and in using a multiply paral-lel test set, as in 2017, it was unusual also in fea-turing an unprecedented number of languages and treebanks and in integrating cross-lingual learning for resource-poor languages. Compared to the first edition of the task in 2017, this year several lan-guages were provided with little-to-no resources, whereas in 2017, predicted morphology trained on

¹⁵This is true of at least 3 of the 5 best performing systems.

¹⁶The baseline embeddings were the same as in 2017 and therefore did not cover new languages, which may partly ex-plain the greater popularity of the Facebook embeddings this year.

¹⁷We know that some teams used them also for clusters involving high-resource languages, but we have no detailed statistics on this usage.

the language in question was available for all of the languages. The most extreme example of these is Thai, where the only accessible resource was the Facebook Research Thai embeddings model and the OPUS parallel corpora. This year’s task also introduced two additional metrics that take into account morphology and lemmatization. This en-couraged the development of truly end-to-end full parsers, producing complete parses including mor-phological features and lemmas in addition to the syntactic tree. This also aimed to improve the utili-ty of the systems developed in the shared task for later downstream applications. For most UD lan-guages, these parsers represent a new state of the art for end-to-end dependency parsing.

The analysis of the shared task results has so far only scratched the surface, and we refer to the sys-tem description papers for more in-depth analysis of individual systems and their performance. For many previous CoNLL shared tasks, the task it-self has only been the starting point of a long and fruitful research strand, enabled by the resources created for the task. We hope and believe that the 2017 and 2018 UD parsing tasks will join this tra-dition.

Acknowledgments

We are grateful to all the contributors to Universal Dependencies; without their effort a task like this simply wouldn’t be possible.

The work described herein, including data preparation for the *CoNLL 2018 UD Shared Task*, has been supported by the fol-lowing grants and projects: OP PPR No. CZ.07.1.02/0.0/0.0/16.023/0000108 of the City of Prague, “CRACKER,” H2020 Project No. 645357 of the European Commission; “MANYLA,” Grant No. GA15-10472S of the Grant Agency of the Czech Republic; FIN-CLARIN; Grant No. 2016-01817 of the Swedish Research Council; and the LINDAT/CLARIN research infrastructure project funded by the Ministry of Education, Youth and Sports of the Czech Republic, Project. No. LM2015071. The data for the *CoNLL 2018 UD Shared Task* are available also via the LINDAT/CLARIN repository.

References

Gor Arakelyan, Karen Hambardzumyan, and Hrant Khachatryan. 2018. Towards JointUD: Part-of-speech tagging and lemmatization using recurrent

- neural networks. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Riyaz Ahmad Bhat, Irshad Ahmad Bhat, and Srinivas Bangalore. 2018. The SLT-Interactions parsing system at the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information *arXiv preprint arXiv:1607.04606*.
- Tiberiu Boros, Stefan Daniel Dumitrescu, and Ruxandra Burtica. 2018. NLP-Cube: End-to-end raw text processing with neural networks. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Sabine Buchholz and Erwin Marsi. 2006. [CoNLL-X shared task on multilingual dependency parsing](#). In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*. Association for Computational Linguistics, pages 149–164. <http://anthology.aclweb.org/W/W06/W06-29.pdf#page=165>.
- Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Danlu Chen, Mengxiao Lin, Zhifeng Hu, and Xipeng Qiu. 2018. A simple yet effective joint training method for cross-lingual universal dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Elie Duthoo and Olivier Mesnard. 2018. CEA LIST : Processing low-resource languages for CoNLL 2018. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Murhaf Fares, Stephan Oepen, Lilja Øvrelid, Jari Björne, and Richard Johansson. 2018. The 2018 shared task on extrinsic parser evaluation: On the downstream utility of English universal dependency parsers. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, Brussels, Belgium.
- Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. [CoNLL 2017 shared task - automatically annotated raw texts and word embeddings](#). LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University. <http://hdl.handle.net/11234/1-1989>.
- Allan Hanbury, Henning Müller, Krisztian Balog, Torben Brodt, Gordon V. Cormack, Ivan Eggel, Tim Gollub, Frank Hopfgartner, Jayashree Kalpathy-Cramer, Noriko Kando, Anastasia Krithara, Jimmy Lin, Simon Mercer, and Martin Potthast. 2015. [Evaluation-as-a-Service: Overview and Outlook](#). *ArXiv e-prints* <http://arxiv.org/abs/1512.07454>.
- Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018. Universal dependency parsing with a general transition-based DAG parser. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Ganesh Jawahar, Benjamin Muller, Amal Fethi, Louis Martin, Éric de La Clergerie, Benoît Sagot, and Djamé Seddah. 2018. ELMoLex: Connecting ELMo and lexicon features for dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Tao Ji, Yufang Liu, Yijun Wang, Yuanbin Wu, and Man Lan. 2018. AntNLP at CoNLL 2018 shared task: A graph-based parser for universal dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku neural parser pipeline: An end-to-end system for the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Ömer Kirnap, Erenay Dayanık, and Deniz Yuret. 2018. Tree-stack LSTM in transition based dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Zuchao Li, Shexia He, Zhuosheng Zhang, and Hai Zhao. 2018. Joint learning of pos and dependencies for multilingual universal dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- KyungTae Lim, Cheoneum Park, Changki Lee, and Thierry Poibeau. 2018. SEx BiST: A multi-source trainable parser with deep contextualized lexical representations. In *Proceedings of the CoNLL 2018*

Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Association for Computational Linguistics.

Dat Quoc Nguyen and Karin Verspoor. 2018. An improved neural network model for joint pos tagging and dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, John Bauer, Sandra Bellato, Kepa Bengoetxea, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Tomáš Jelínek, Anders Johansen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskiy,

Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horfiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayò Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Siyao Peng, Ceneil-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Riebler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roşca, Olga Rudina, Shoval Sadde, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Šimi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdenka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. *Universal dependencies 2.2*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. <http://hdl.handle.net/11234/1-2837>.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. *Universal Dependencies v1: A multilingual treebank collection*. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, Portorož, Slovenia, pages 1659–1666. <http://www.lrec-conf.org/proceedings/lrec2016/summaries/348.html>.

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.

- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. [The CoNLL 2007 shared task on dependency parsing](#). In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*. Association for Computational Linguistics, pages 915–932. <http://www.aclweb.org/anthology/D/D07/D07-1.pdf#page=949>.
- Berkay Furkan Önder, Can Gümeli, and Deniz Yuret. 2018. SParse: Koç University graph-based parsing system for the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Şaziye Betül Özateş, Arzucan Özgür, Tunga Güngör, and Balkız Öztürk. 2018. A morphology-based representation model for LSTM-based dependency parsing of agglutinative languages. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. 2017. [Udapi: Universal API for universal dependencies](#). In *NoDaLiDa 2017 Workshop on Universal Dependencies*. Göteborgs universitet, Göteborg, Sweden, pages 96–101. <http://aclweb.org/anthology/W/W17/W17-0412.pdf>.
- Martin Potthast, Tim Gollub, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. 2014. [Improving the reproducibility of PAN’s shared tasks: Plagiarism detection, author identification, and author profiling](#). In Evangelos Kanoulas, Mihai Lupu, Paul Clough, Mark Sanderson, Mark Hall, Allan Hanbury, and Elaine Toms, editors, *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14)*. Springer, Berlin Heidelberg New York, pages 268–299. https://doi.org/10.1007/978-3-319-11382-1_22.
- Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Rudolf Rosa and David Mareček. 2018. CUNI x-ling: Parsing under-resourced languages in CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Piotr Rybak and Alina Wróblewska. 2018. Semi-supervised neural system for tagging, parsing and lematization. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Amit Seker, Amir More, and Reut Tsarfaty. 2018. Universal morpho-syntactic parsing and the contribution of lexica: Analyzing the ONLP submission to the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 treebanks, 34 models: Universal dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- 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*. Association for Computational Linguistics.
- Milan Straka, Jan Hajič, and Jana Straková. 2016. UD-Pipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, Portorož, Slovenia.
- 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*. Association for Computational Linguistics.
- Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)*. European Language Resources Association (ELRA), Istanbul, Turkey.
- Hui Wan, Tahira Naseem, Young-Suk Lee, Vittorio Castelli, and Miguel Ballesteros. 2018. IBM Research at the CoNLL 2018 shared task on multilingual parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Yingting Wu, Hai Zhao, and Jia-Jun Tong. 2018. Multilingual universal dependency parsing from raw text with low-resource language enhancement. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gökırmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Lung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkor-eit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadova, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. [Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies](http://www.aclweb.org/anthology/K17-3001). In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, Vancouver, Canada, pages 1–19. <http://www.aclweb.org/anthology/K17-3001>.

6.6 Towards Deep Universal Dependencies

Full reference: Kira Droganova and Daniel Zeman. Towards Deep Universal Dependencies. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, pages 144–152, Paris, France, August 2019. Association for Computational Linguistics. DOI 10.18653/v1/W19-7717. URL <https://aclanthology.org/W19-7717.pdf>. [Droganova and Zeman, 2019]

Comments: This paper is the first step on the journey from Universal Dependencies to a similarly broad and multilingual approach to deep syntax and semantics. As such, it is a representative of the possible future directions I outline in Chapter 5. We have already released several automatic enhancements of Universal Dependencies with deep-syntactic annotations and received some feedback from other researchers. Nevertheless, Deep UD has to be considered work in progress: it will be really useful when it can incorporate existing manually curated resources such as Prague tectogrammar or PropBank. My contribution: 50%. Number of citations according to Google Scholar (retrieved 2023-07-21): **16**.

Towards Deep Universal Dependencies

Kira Drohanova

Charles University

Faculty of Mathematics and Physics

Praha, Czechia

drohanova@ufal.mff.cuni.cz

Daniel Zeman

Charles University

Faculty of Mathematics and Physics

Praha, Czechia

zeman@ufal.mff.cuni.cz

Abstract

Many linguistic theories and annotation frameworks contain a deep-syntactic and/or semantic layer. While many of these frameworks have been applied to more than one language, none of them is anywhere near the number of languages that are covered in Universal Dependencies (UD). In this paper, we present a prototype of Deep Universal Dependencies, a two-speed concept where minimal deep annotation can be derived automatically from surface UD trees, while richer annotation can be added for datasets where appropriate resources are available. We release the Deep UD data in Lindat.

1 Introduction

Universal Dependencies (UD) (Nivre et al., 2016) annotation guidelines have become a de-facto standard for cross-linguistically comparable morphological and syntactic annotation. A significant factor in the popularity of UD is a steadily growing and heavily multilingual collection of corpora: release 2.4 (Nivre et al., 2019) contains 146 treebanks of 83 languages. The UD guidelines have been designed as surface-syntactic, although their emphasis on cross-linguistic parallelism sometimes leads to decisions that are normally associated with deeper, semantics-oriented frameworks (the primacy of content words and the second-class citizenship of function words may serve as an example).

Many theories and annotation frameworks have been proposed that contain a deep-syntactic, teogrammatical, or semantic dependency layer; to name just a few: Meaning-Text Theory (Žolkovskij and Mel’čuk, 1965), Functional Generative Description (Sgall, 1967), the Proposition Bank (Kingsbury and Palmer, 2002), Sequoia (Candito and Seddah, 2012), or Abstract Meaning Representation (Banarescu et al., 2013). Names vary and so does the extent of ‘deep’ phenomena that are annotated; the common denominator is that these phenomena are closer to meaning on the meaning-form scale than anything we find in a typical surface-syntactic treebank. By definition, deep representation is more useful for natural language understanding (but it is also more difficult to obtain).

Many of the deep frameworks have been applied to more than one language, sometimes just to demonstrate that it is possible; but none of them is anywhere near the number of languages covered by UD.

UD itself contains a diffident attempt to provide deeper annotations, dubbed the Enhanced Universal Dependencies (Schuster and Manning, 2016). While it is a step in the right direction, it is just the first step: we argue that it should be possible to go deeper. Moreover, Enhanced UD is an optional extension, which is only available in a handful of treebanks (Table 1). Enhanced UD faces the same threat as the other deep frameworks mentioned above: more complex annotation requires more annotation effort, and semantic annotations are often coupled with huge lexical resources such as verb frame dictionaries. Therefore, it is less likely that sufficient manpower will be available to annotate data in a new language. Our principal question is thus the following: is it possible to create a multilingual data collection (and annotation guidelines) that will be as popular and widely used as UD, but deeper?

In our view, the key is to identify a subset of deep annotations that can be derived semi-automatically from surface UD trees, in acceptable quality. These annotations will not be as precise as if they were carefully checked by humans, but they will be available for (almost) all UD languages. More importantly, it will be possible to generate them for new UD languages and the deep extension will thus keep up with

the growth of UD. For languages that have better resources available, one could convert them to the deep UD format and provide them instead of the corresponding semi-automatic annotation. Note that there are two dimensions along which a resource can be ‘better’. It can provide the same type of annotation as the light, semi-automatic version, just verified by human annotators. But it may also provide additional types of annotations that cannot be obtained automatically. The Deep UD guidelines should thus cover a broad selection of phenomena that are annotated in popular semantic dependency frameworks.

The present paper reports on work in progress. We have prepared the first prototype of the semi-automatic Deep Universal Dependencies, based on UD release 2.4. The resource is available in the LINDAT/CLARIN repository (<http://hdl.handle.net/11234/1-3022>) under the same set of licenses as the underlying UD treebanks. In the following sections we describe what types of annotation this first version contains and how the annotation is derived from the surface trees; we also offer an outlook on possible future development.

2 Related Work

Manual semantic annotation is a highly time-consuming process, therefore a number of authors experimented with (semi-)automatic approaches to semantic annotation. Padó (2007) proposed a method that uses parallel corpora to project annotation to transfer semantic roles from English to resource-poorer languages. The experiment was conducted on an English-German corpus. Van der Plas et al. (2011) experimented with joint syntactic-semantic learning aiming at improving the quality of semantic annotations from automatic cross-lingual transfer. An alternative approach was proposed by Exner et al. (2016). Instead of utilizing parallel corpora, they use loosely parallel corpora where sentences are not required to be exact translations of each other. Semantic annotations are transferred from one language to another using sentences aligned by entities. The experiment was conducted using the English, Swedish, and French editions of Wikipedia. Akbik et al. (2015) described a two-stage approach to cross-lingual semantic role labeling (SRL) that was used to generate Proposition Banks for 7 languages. First, they applied a filtered annotation projection to parallel corpora, which was intended to achieve higher precision for a target corpus, even if containing fewer labels. Then they bootstrapped and retrained the SRL to iteratively improve recall without reducing precision. This approach was also applied to 7 treebanks from UD release 1.4.¹ However, the project seems to be stalled.

Mille et al. (2018) proposed the deep datasets that were used in the Shallow and Deep Tracks of the Multilingual Surface Realisation Shared Task (SR’18, SR’19). The Shallow Track datasets consist of unordered syntactic trees with all the word forms replaced with their lemmas; part-of-speech tags and the morphological information are preserved (available for 10 languages). The Deep Track datasets consist of trees that contain only content words linked by predicate-argument edges in the PropBank fashion (available for English, French and Spanish). The datasets were automatically derived from UD trees v.2.0. Gotham and Haug (2018) proposed an approach to deriving semantic representations from UD structures that is based on techniques developed for Glue semantics for LFG. The important feature of this approach is that it relies on language-specific resources as little as possible.

3 Enhanced Universal Dependencies

The Enhanced UD (Schuster and Manning, 2016)² represents a natural point of departure for us. UD v2 guidelines define five types of enhancements that can appear in treebanks released as part of UD. All the enhancements are optional and it is possible for a treebank to annotate one enhancement while ignoring the others. The enhanced representation is a directed graph but not necessarily a tree. It may contain ‘null’ nodes, multiple incoming edges and even cycles. The following enhancements are defined:

¹<https://github.com/System-T/UniversalPropositions>

²While Schuster and Manning (2016) remains the most suitable reference for Enhanced UD to date, its publication predates the v2 UD guidelines and the proposals it contains are only partially compliant with the guidelines. See <https://universaldependencies.org/u-overview/enhanced-syntax.html> for the current version.

Null nodes for elided predicates. In certain types of ellipsis (*gapping* and *stripping*), multiple copies of a predicate are understood, each with its own set of arguments and adjuncts, but only one copy is present on the surface. Example: *Mary flies to Berlin and Jeremy [flies] to Paris*. The enhanced graph contains an extra node for each copy of the predicate that is missing on the surface. Note that the guidelines do not license null nodes for other instances of ellipsis, such as dropped subject pronouns in pro-drop languages.

Propagation of conjuncts. Coordination groups several constituents that together play one role in the superordinate structure. They are all equal, despite the fact that the first conjunct is formally treated as the head in the basic UD tree. For example, several coordinate nominals may act as subjects of a verb, but only the first nominal is actually connected with the verb via an *nsubj* relation. In the enhanced graph, this relation is propagated to the other conjuncts, i.e., each coordinate nominal is directly connected to the verb (in addition to the *conj* relation that connects it to the first conjunct). Likewise, there may be shared dependents that are attached to the first conjunct in the basic tree, but in fact they modify the entire coordination. Their attachment will be propagated to the other conjuncts, too. (Note that not all dependents of the first conjunct must be shared. Some of them may modify only the first conjunct, especially if the other conjuncts have similar dependents of their own.)

External subjects. Certain types of non-finite, ‘open’ clausal complements inherit their subject from the subject or the object of the matrix clause. Example: *Susan wants to buy a book*. In the basic tree, *Susan* will be attached as *nsubj* of *wants*, while there will be no subject dependent of *buy*. In contrast, the enhanced graph will have an additional *nsubj* relation between *buy* and *Susan*.

Relative clauses. The noun modified by a relative clause plays a semantic role in the frame of the subordinate predicate. In the basic UD tree, it is represented by a relative pronoun; however, in the enhanced graph it is linked from the subordinate predicate *instead* of the pronoun. (The pronoun is detached from the predicate and attached to the noun it represents, via a special relation *ref*.) This is the reason why enhanced graphs may contain cycles: in *The boy who lived*, there is an *acl:relcl* relation from *boy* to *lived*, and an *nsubj* relation from *lived* to *boy*.

Case information. The labels of certain dependency relations are augmented with case information, which may be an adposition, a morphological feature, or both. For example, the German prepositional phrase *auf dem Boden* “on the ground” may be attached as an oblique dependent (*obl*) of a verb in the basic tree. The enhanced label will be *obl:auf:dat*, reflecting that the phrase is in the dative case with the preposition *auf*. This information is potentially useful for semantic role disambiguation, and putting it to the label is supposed to make it more visible; nevertheless, its acquisition from the basic tree is completely deterministic, and there is no attempt to translate the labels to a language-independent description of meaning.

Several extensions of the enhanced representation have been proposed. The *enhanced++* graphs proposed by Schuster and Manning (2016) extend the set of ellipsis-in-coordination types where null nodes are added; they also suppress quantifying expressions in sentences like *a bunch of people are coming*.

Candito et al. (2017) define the *enhanced-alt* graphs, which neutralize syntactic alternations, that is, passives, medio-passives, impersonal constructions and causatives. They also suggest to annotate external arguments of other non-finite verb forms than just open infinitival complements and relative clauses: most notably, for participles, even if they are used attributively. Hence in *ceux embauchés en 2007* “those hired in 2007”, *embauchés* heads a non-relative adnominal clause (*acl*) that modifies the nominal *ceux*, but at the same time *ceux* is attached as a passive subject (*nsubj:pass*) of *embauchés*.

4 Pre-existing Enhancing Tools

Enhanced UD contains information that cannot be derived automatically from the basic UD tree; additional human input is needed in order to fully disambiguate all situations. Nevertheless, it is believed that automatic ‘enhancers’ can get us relatively far. Schuster and Manning (2016) described and evaluated

the Stanford Enhancer,³ which is available as a part of the Stanford CoreNLP suite.

Nyblom et al. (2013) reported on the Turku Enhancer, a hybrid approach (consisting of rule-based heuristics and machine-learning components) to enhancing Stanford Dependencies of Finnish. The enhancements tackled were conjunct propagation, external subjects, and syntactic functions of relativizers; the first two are thus relevant also in Enhanced UD. Their system achieved F_1 score of 93.1; note however that labeled training data is needed for the approach to work.

Nivre et al. (2018) compares the Stanford Enhancer with an adapted version of the Turku Enhancer. They trained it on the Finnish labeled data, but in a delexicalized fashion (only non-lexical features were considered). The Turku Enhancer does not predict null nodes, and for external subjects it only considers subject control (or raising), but not object control. On the other hand, Stanford Enhancer only predicts core arguments as controllers while in some languages non-core dependents can control subjects too. Nevertheless, both enhancers are found usable for other languages, as shown on Swedish and Italian. The paper also evaluates an Italian-specific rule-based enhancer, which does not predict null nodes.

Candito et al. (2017) took a rule-based approach to produce their *enhanced-alt* graphs for French: they developed two sets of rules, using two different graph rewriting systems. However, they only focus on two of the five enhancements (external subjects and conjunct propagation), and they only do it for French. Some of their heuristics are very French-specific and they assume that information needed for disambiguation is available in the source annotation (which is the case of the Sequoia French treebank).

Several other UD treebanks come from sources where some enhanced annotation is available and can be converted to Enhanced UD. Bouma (2018) demonstrates how original annotations from the Alpino treebank can help enhance the Dutch UD treebanks. Patejuk and Przepiórkowski (2018) discuss conversion from an LFG treebank of Polish and note that not only there is more information than in basic UD, some information cannot be captured even by Enhanced UD. Another example is the distinction between private and shared dependents in coordination: for treebanks converted from Prague-style annotation (Arabic, Czech, Lithuanian, Slovak, Tamil), this distinction is readily available.

5 Data Preparation

The first version of Deep UD is based on UD release 2.4 (Nivre et al., 2019) but we intend to generate updates after each future UD release. While we foresee improved semantic annotation for some languages (based on additional lexical resources, for example), the current version is derived just from the annotation available in UD itself (though we use heuristics that may be language- or treebank-specific). UD 2.4 contains 146 treebanks of 83 languages. We exclude 6 treebanks that are distributed, for copyright reasons, as hollow annotations without the underlying text. We further exclude 19 treebanks with incomplete or non-existent lemmatization.⁴ Consequently, our resource contains 121 treebanks of 73 languages.

We take enhanced UD graphs (Section 3) as the point of departure for deep UD. However, only a small fraction of the UD treebanks have some enhanced annotation, and if they do, then they often omit one or more of the five types of enhancements defined in the guidelines. There are 24 treebanks of 16 languages that have enhanced graphs (Table 1). We will refer to these enhanced graphs as *trusted enhanced annotations*. Some of them were converted from non-UD manual annotations, some were probably generated with the help of automatic enhancers, but at least they were overseen by the teams responsible for the given language.

We use the Stanford Enhancer⁵ to generate enhanced graphs for corpora that lack them. For the six treebanks in Table 1 that contain trusted annotation of all five enhancement types, we take the trusted annotation. For the other 18 treebanks in the table, ideally we should merge the trusted annotation with the output of the enhancer so that all enhancement types are present. However, merging may not be trivial

³The Stanford UD Enhancer was adapted from an older tool that was designed to work with the Stanford Dependencies, a predecessor of UD.

⁴Note that we do not exclude some other treebanks where lemmas exist but have been assigned by a stochastic model instead of human annotators.

⁵The README file of the released data provides details on what version we used and how we ran it.

Language	Treebank	Gapping	Coord	XSubj	RelCl	CaseDeprel
Arabic	PADT		yes			
Bulgarian	BTB		yes	yes	yes	yes
Czech	CAC		yes			
Czech	FicTree		yes			
Czech	PDT		yes			
Dutch	Alpino	yes	yes	yes	yes	yes
Dutch	LassySmall	yes	yes	yes	yes	yes
English	EWT	yes	yes	yes	yes	yes
English	PUD	yes	yes	yes	yes	yes
Estonian	EWT	yes				
Finnish	PUD	yes	yes			
Finnish	TDT	yes	yes	yes		
Italian	ISDT		yes	yes	yes	yes
Latvian	LVTB	yes	yes	yes		yes
Lithuanian	ALKSNIS		yes			
Polish	LFG		yes	yes		yes
Polish	PDB		yes			
Polish	PUD		yes			
Russian	SynTagRus	yes				
Slovak	SNK		yes			
Swedish	PUD	yes	yes	yes	yes	yes
Swedish	Talbanken	yes	yes	yes	yes	yes
Tamil	TTB		yes			
Ukrainian	IU	yes	yes	yes	yes	

Table 1: Overview of enhanced annotations in UD 2.4 treebanks. Gapping: there are empty nodes representing elided predicates. Coord: dependencies (both incoming and outgoing) are propagated to all conjuncts. XSubj: higher argument is linked as the subject of a controlled verb. RelCl: nominal modified by a relative clause is linked as argument or adjunct in that clause. CaseDeprel: case markers are added to the dependency labels of adverbial and oblique dependents.

in sentences where multiple enhancement types interact, and we leave it for future work. In the current version, the enhanced graphs in these 18 treebanks are replaced by the output of the Stanford Enhancer.

Note that using the Stanford Enhancer does not guarantee that the resulting annotation identifies all five types of enhancements—even if the phenomenon exists in the language and the treebank is large enough to provide examples. Identification of relative clauses relies on a language-specific list of relative pronouns and on the optional dependency label `acl:relcl`, but some treebanks use `acl` instead. Gapping, besides being relatively rare, is not annotated properly in the basic representation of some UD languages. Consequently, only 58 enhanced treebanks have some null nodes (gapping) and only 54 treebanks have edges specific to relative-clause enhancements. Most treebanks have the other three types; a remarkable exception is Japanese where the three treebanks have only one enhancement type, namely the case-augmented dependency relations. 37 treebanks feature all five types. We plan to expand the relative clause annotation to other treebanks in the future; listing relative pronouns (a closed class) is quite feasible, and we can utilize the morphological feature `PronType=Rel` where available.

6 Delving Deeper

There are numerous phenomena that various semantic frameworks strive to capture. Without precluding any of them from future versions of Deep UD, we believe that the core of sentence understanding is its predicate-argument structure. We start with verbal predicates and identify their arguments, if present in the same sentence. We number the arguments roughly reflecting their decreasing salience and making

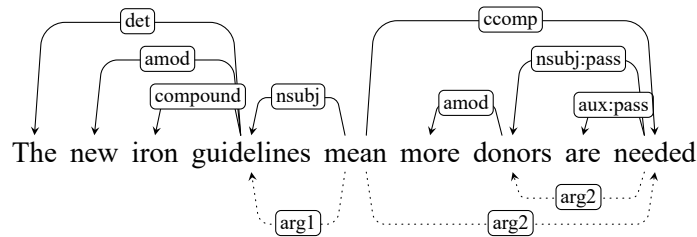


Figure 1: An example of a deep graph for the English sentence *The new iron guidelines mean more donors are needed*.

sure that for the same predicate (sense), the argument with a particular semantic role will always get the same label/number, regardless the syntactic environment. That means that we have to neutralize valency-changing operations such as passivization; here we are very close to the *enhanced-alt* representation proposed by Candito et al. (2017). For example, in *George killed the dragon* as well as in *The dragon was killed by George*, *George* will be *arg1* and *the dragon* will be *arg2*. We do not label the actual semantic roles (i.e., agent / actor / killer for *George* and patient / killed for *the dragon*) directly in the text. Instead, the predicate instance can be linked to a frame dictionary (if available) where the corresponding frame will provide interpretation of the numbered arguments. Linking of frame instances to dictionary frames will not be trivial and the concrete approach will depend on the language and on the nature of the target lexical resource. Valency frame dictionaries often contain information on morphological and syntactic properties of the arguments. A verbal lemma will typically correspond to several (sometimes dozens of) different frames. Sometimes the forms of the arguments (their morphological case, preposition etc.) will narrow down the search; but full disambiguation may not be possible without a statistical model or a human annotator. Once we have the correct frame, identification of individual arguments is (again) just matching their properties against those specified by the frame.

We follow the CoNLL-U Plus file format⁶ with two new columns: DEEP:PRED and DEEP:ARGS. These columns contain annotation we add on top of Enhanced UD; without them, the file is still a valid CoNLL-U file. The value in DEEP:PRED identifies the predicate. It can be a reference to a particular sense (frame) in a dictionary but we currently use just the lemma of the verb, possibly augmented with other lemmas if it is a compound verb (e.g. Germanic phrasal verbs such as *come up*). The value in DEEP:ARGS points to the head nodes of subtrees that represent the arguments. For example, *arg1 : 33 | arg2 : 12, 27* means that the most salient argument (possibly the agent) is headed by node 33, while the second most salient argument (possibly the patient) is coordination and the conjuncts are headed by nodes 12 and 27, respectively. See Figure 1 for an example of a deep graph.

Thanks to Enhanced UD, the annotation resolves some instances of grammatical coreference (Zikánová et al., 2015), i.e., situations where one node serves as an argument of multiple verbs, and it can be inferred from the grammatical rules of the language. On the other hand, the current version does not attempt to address textual coreference, e.g., a personal pronoun that is coreferential with a noun. Arguably, textual coreference cannot be resolved without a human annotator or a trained model.

Some arguments are not accessible through Enhanced UD; similar to Candito et al. (2017), we are experimenting with heuristics that yield additional enhanced dependencies for non-finite verbs:

Infinitives that are not *xcomp*. They can be ordinary clausal complements (*ccomp*) and then we cannot identify their subject, as in Dutch: *Zijlaard adviseerde te gokken op de sprint* (lit. *Zijlaard advised to bet on the sprint*) “Zijlaard advised betting on the sprint”. But they can be also adverbial clauses (*advcl*), or adnominal clauses (*ac1*), if the main clause’s predicate is a light verb with a noun, as in Dutch: *had moeite om zich te concentreren* (lit. *had trouble so himself to concentrate*) “struggled to concentrate”. The infinitive *concentreren* “to concentrate” in this case works similarly to an *xcomp*, that is, it should inherit the subject from the matrix clause.

⁶<https://universaldependencies.org/ext-format.html>

Participles. An attributively used participle modifies a noun. If it were a relative clause, the enhanced graph would identify the noun as the “subject” argument of the participle; but it is an amod rather than a clause, and no external subject relation is present. A Dutch example: *de afgelopen week* (lit. *the expired week*) “last week”. We add a heuristic that participles attached as amod shall take the modified noun as their argument; note that we need to distinguish active and passive participles in order to find out whether the noun is argument 1 or 2. Currently we only look for the morphological feature `Voice=Pass` but it is not always available, and some verb forms can be used both in active and passive clauses. Consider English: *the shares reflected on your statement*; *reflected* is used as a passive participle but `Voice=Pass` is not present, it is just a “past participle” without any voice feature. We may need to estimate whether a verb is transitive, and if it is, the participle will be considered passive, otherwise it will be considered active. Nevertheless, no such heuristic was applied to the current version of the data.

Converbs (gerunds). English: *X did Y..., killing several people*. The syntactic annotation does not tell us that X is the argument 1 of *killing*. We work with the hypothesis that a gerund or converb attached as `advcl` inherits the subject of the matrix clause. This is a rule at least in some languages but we have yet to evaluate to what extent the rule may be universal.

Language-specific heuristics. A number of heuristics will be needed that are language- or even treebank-specific. For example, passivization of English ditransitive clauses promotes the indirect object rather than the direct object (*what I was asked*).⁷ Therefore, if there is a direct object in a passive clause, the subject should be considered argument 3 and not 2.

7 Conclusion and Outlook

We presented a prototype of Deep Universal Dependencies, a deep-syntactic annotation layer that can be derived semi-automatically from surface UD graphs. Our plan is to accommodate rich semantic annotations in languages where necessary resources are available, and automatically generate the core part for other languages after each UD release. Our contribution at the current stage is threefold: 1. While UD releases still contain Enhanced UD only for a few treebanks, we make sure that enhanced graphs are available everywhere; 2. to find more arguments, we do additional enhancements (infinitives, gerunds, participles) internally but we do not show them in the enhanced graphs so that the graphs stay within the current guidelines; 3. we normalize diathesis and show the numbered arguments (canonical subject and object in the terms of Candito et al. (2017)).

The list of possible future directions is much longer than we can accommodate in a short paper; for instance, we want to take advantage of oblique argument marking in treebanks where it is available, improve recognition of passives and other diathesis alternations, or implement other enhancements from Schuster and Manning (2016)’s *enhanced++*. Nevertheless, the most important next step is to evaluate the quality of the generated annotation (both the output of the Stanford Enhancer and the additional heuristics we applied to the enhanced graphs). Since there is no gold-standard labeled data suitable for such evaluation, we will have to manually inspect random samples of the output, or compare the predicate-argument patterns with existing valency dictionaries (in languages where they exist).

Acknowledgements

This work is partially supported by the GA UK grant 794417, the SVV project number 260 453, and the grant no. LM2015071 of the Ministry of Education, Youth and Sports of the Czech Republic.

We also wish to thank the three anonymous reviewers for their valuable comments.

References

Alan Akbik, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, Huaiyu Zhu, et al. 2015. Generating high quality proposition banks for multilingual semantic role labeling. In Proceedings of the 53rd Annual Meeting of

⁷Of course, one could then question whether *I* is an indirect object in the active clause if it can be promoted by passivization; here we follow the actual approach of the English UD treebanks.

the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), volume 1, pages 397–407.

- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, pages 178–186.
- Gosse Bouma. 2018. Comparing two methods for adding enhanced dependencies to UD treebanks. In Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018), pages 17–30, Oslo, Norway. Linköping Electronic Conference Proceedings.
- Marie Candito and Djamé Seddah. 2012. Le corpus Sequoia: annotation syntaxique et exploitation pour l’adaptation d’analyseur par pont lexical. In TALN 2012-19e conférence sur le Traitement Automatique des Langues Naturelles.
- Marie Candito, Bruno Guillaume, Guy Perrier, and Djamé Seddah. 2017. Enhanced UD dependencies with neutralized diathesis alternation. In Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017), pages 42–53, Pisa, Italy.
- Peter Exner, Marcus Klang, and Pierre Nugues. 2016. Multilingual supervision of semantic annotation. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1007–1017.
- Matthew Gotham and Dag Trygve Truslew Haug. 2018. Glue semantics for Universal Dependencies. In Miriam Butt and Tracy Holloway King, editors, Proceedings of the LFG’18 Conference, pages 208–226, Wien, Austria. CSLI Publications.
- Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In LREC, pages 1989–1993.
- Simon Mille, Anja Belz, Bernd Bohnet, and Leo Wanner. 2018. Underspecified universal dependency structures as inputs for multilingual surface realisation. In Proceedings of the 11th International Conference on Natural Language Generation, pages 199–209.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), pages 1659–1666, Paris, France. European Language Resources Association.
- Joakim Nivre, Paola Marongiu, Filip Ginter, Jenna Kanerva, Simonetta Montemagni, Sebastian Schuster, and Maria Simi. 2018. Enhancing universal dependency treebanks: A case study. In Proceedings of the Second Workshop on Universal Dependencies (UDW 2018), pages 102–107, Bruxelles, Belgium. Association for Computational Linguistics.
- Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Gabrielė Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabrizio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaz Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos García, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökirmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Matias Gironi, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Oľájdé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Andre Kaasen, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Arne Köhn, Kamil Kopacewicz, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev,

- John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Yuan Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horňáček, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayò Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Lapińska, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Jack Rueter, Shoal Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Sarah McGuinness, Abigail Walsh, Dage Sörg, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamel Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, et al. 2019. Universal dependencies 2.4. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Jenna Nyblom, Samuel Kohonen, Katri Haverinen, Tapio Salakoski, and Filip Ginter. 2013. Predicting conjunct propagation and other extended Stanford Dependencies. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 252–261, Praha, Czechia. Matfyzpress.
- Sebastian Padó. 2007. *Cross-lingual annotation projection models for role-semantic information*. Saarland University.
- Agnieszka Patejuk and Adam Przepiórkowski. 2018. *From Lexical Functional Grammar to Enhanced Universal Dependencies*. Instytut Podstaw Informatyki Polskiej Akademii Nauk, Warszawa, Poland.
- Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association.
- Petr Sgall. 1967. Functional sentence perspective in a generative description. *Prague Studies in Mathematical Linguistics*, 2:203–225.
- Lonneke Van der Plas, Paola Merlo, and James Henderson. 2011. Scaling up automatic cross-lingual semantic role annotation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 299–304. Association for Computational Linguistics.
- Šárka Zikánová, Eva Hajičová, Barbora Hladká, Pavlína Jínová, Jiří Mírovský, Anna Nedoluzhko, Lucie Poláková, Kateřina Rysová, Magdaléna Rysová, and Jan Václ. 2015. *Discourse and Coherence. From the Sentence Structure to Relations in Text*. Studies in Computational and Theoretical Linguistics. ÚFAL, Praha, Czechia.
- Aleksandr K. Žolkovskij and Igor A. Mel'čuk. 1965. O vozmožnom metode i instrumentax semantičeskogo sinteza (on a possible method and instruments for semantic synthesis). *Naučno-texničeskaja informacija*, 5:23–28.

Bibliography

- Željko Agić, Dirk Hovy, and Anders Søgaard. If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 268–272, Beijing, China, July 2015. Association for Computational Linguistics. URL <https://aclanthology.org/P15-2044.pdf>.
- Ika Alfina, Daniel Zeman, Arawinda Dinakaramani, Indra Budi, and Heru Suhartanto. Selecting the UD v2 morphological features for Indonesian dependency treebank. In *Proceedings of the International Conference on Asian Language Processing (IALP 2020)*, pages 104–109, Kuala Lumpur, Malaysia, 2020. Chinese and Oriental Languages Information Processing Society. ISBN 978-1-7281-7689-5. URL https://colips.org/conferences/ialp2020/proceedings/papers/IALP2020_P87.pdf.
- Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. Zero-resource dependency parsing: Boosting delexicalized cross-lingual transfer with linguistic knowledge. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 119–130, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL <https://aclanthology.org/C16-1012.pdf>.
- Gosse Bouma, Djamé Seddah, and Daniel Zeman. Overview of the IWPT 2020 shared task on parsing into enhanced Universal Dependencies. In Gosse Bouma, Yuji Matsumoto, Stephan Oepen, Kenji Sagae, Djamé Seddah, Weiwei Sun, Anders Søgaard, Reut Tsarfaty, and Daniel Zeman, editors, *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 151–161, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.iwpt-1.16. URL <https://aclanthology.org/2020.iwpt-1.16>.
- Gosse Bouma, Djamé Seddah, and Daniel Zeman. From raw text to enhanced Universal Dependencies: The parsing shared task at IWPT 2021. In Stephan Oepen, Kenji Sagae, Reut Tsarfaty, Gosse Bouma, Djamé Seddah, and Daniel Zeman, editors, *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 146–157, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.iwpt-1.15. URL <https://aclanthology.org/2021.iwpt-1.15>.
- Sabine Buchholz and Erwin Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York, NY, USA, June 2006. Association for Computational Linguistics. URL <https://aclanthology.org/W06-2920.pdf>.
- Marie Candito, Bruno Guillaume, Guy Perrier, and Djamé Seddah. Enhanced UD dependencies with neutralized diathesis alternation. In *Proceedings of the*

- Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 42–53, Pisa, Italy, September 2017. Linköping University Electronic Press. URL <https://aclanthology.org/W17-6507.pdf>.
- Flavio Massimiliano Cecchini, Marco Passarotti, Paola Marongiu, and Daniel Zeman. Challenges in converting the Index Thomisticus treebank into Universal Dependencies. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 27–36, Bruxelles, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6004. URL <https://aclanthology.org/W18-6004.pdf>.
- Franck Dary and Alexis Nasr. The Reading Machine: a Versatile Framework for Studying Incremental Parsing Strategies. In *The 17th International Conference on Parsing Technologies*, Bangkok (virtual), Thailand, August 2021. URL <https://hal.science/hal-03328439>.
- Dipanjan Das and Slav Petrov. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 600–609, Portland, OR, USA, June 2011. Association for Computational Linguistics. URL <https://aclanthology.org/P11-1061.pdf>.
- Marie-Catherine de Marneffe and Christopher D. Manning. Stanford typed dependencies manual (technical report), September 2008. URL https://downloads.cs.stanford.edu/nlp/software/dependencies_manual.pdf.
- Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genova, Italy, May 2006. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2006/pdf/440_pdf.pdf.
- Marie-Catherine de Marneffe, Miriam Connor, Natalia Silveira, Samuel R. Bowman, Timothy Dozat, and Christopher D. Manning. More constructions, more genres: Extending Stanford dependencies. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, Praha, Czechia, August 2013. Charles University in Prague, Matfyzpress. URL <https://aclanthology.org/W13-3721.pdf>.
- Marie-Catherine de Marneffe, Natalia Silveira, Timothy Dozat, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, Reykjavík, Iceland, May 2014. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4. URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/1062_Paper.pdf.
- Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, 2021. doi: 10.1162/COLI_a_00402. URL <https://aclanthology.org/2021.cl-2.11.pdf>.

- R. M. W. Dixon. *Basic Linguistic Theory. Volume 1*. Oxford University Press, Oxford, UK, 2010. ISBN 978-0-19-957106-2.
- Kira Droganova and Daniel Zeman. Towards Deep Universal Dependencies. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, pages 144–152, Paris, France, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-7717. URL <https://aclanthology.org/W19-7717.pdf>.
- Kira Droganova, Olga Lyashevskaya, and Daniel Zeman. Data conversion and consistency of monolingual corpora: Russian UD treebanks. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)*, pages 53–66, Linköping, Sweden, 2018. Linköping University Electronic Press. ISBN 978-91-7685-137-1. URL <https://ep.liu.se/ecp/155/ecp18155.pdf#page=61>.
- Puneet Dwivedi and Daniel Zeman. The forest lion and the bull: Morphosyntactic annotation of the Panchatantra. *Computación y Sistemas*, 22(4):1377–1384, 2018. ISSN 1405-5546. URL <https://www.cys.cic.ipn.mx/ojs/index.php/CyS/article/view/3076/2576>.
- Federica Gamba and Daniel Zeman. Universalising Latin Universal Dependencies: a harmonisation of Latin treebanks in UD. In *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*, pages 7–16, Washington, DC, USA, March 2023. Association for Computational Linguistics. URL <https://aclanthology.org/2023.udw-1.2.pdf>.
- Sofia Gustafson-Capková and Britt Hartmann. Manual of the Stockholm Umeå Corpus version 2.0, December 2006. URL <https://spraakbanken.gu.se/parole/Docs/SUC2.0-manual.pdf>. [online; accessed 2018-07-26].
- Jan Hajič, Alena Böhmová, Eva Hajičová, and Barbora Vidová Hladká. The Prague Dependency Treebank: A Three-Level Annotation Scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127. Amsterdam:Kluwer, 2000.
- Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(3):311–325, September 2005. doi: 10.1017/S1351324905003840. URL <https://doi.org/10.1017/S1351324905003840>.
- Olájídé Ishola and Daniel Zeman. Yorùbá dependency treebank (YTB). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5178–5186, Marseille, France, May 2020. European Language Resources Association (ELRA). URL <https://aclanthology.org/2020.lrec-1.637.pdf>.
- Ishan Jindal, Alexandre Rademaker, Michał Ulewicz, Ha Linh, Huyen Nguyen, Khoi-Nguyen Tran, Huaiyu Zhu, and Yunyao Li. Universal proposition bank 2.0. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1700–1711, Marseille, France, June 2022. European Language Resources Association (ELRA). URL <https://aclanthology.org/2022.lrec-1.181.pdf>.

- Dan Kondratyuk and Milan Straka. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1279. URL <https://aclanthology.org/D19-1279.pdf>.
- Matthias T. Kromann. The danish dependency treebank: Linguistic principles and semi-automatic tagging tools, August 2002. URL https://www.researchgate.net/publication/228824624_T_The_Danish_Dependency_Treebank_Linguistic_Principles_and_Semi-automatic_Tagging_Tools.
- Olga Lyashevskaya, Kira Drohanova, Daniel Zeman, Maria Alexeeva, Tatiana Gavrilova, Nina Mustafina, and Elena Shakurova. Universal Dependencies for Russian: A new syntactic dependencies tagset, 2016. URL <http://olesar.narod.ru/papers/44LNG2016.pdf>.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL <https://aclanthology.org/J93-2004.pdf>.
- Héctor Martínez Alonso and Daniel Zeman. Universal Dependencies for the An-Cora treebanks. *Procesamiento del Lenguaje Natural*, 57:91–98, September 2016. URL <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/5341>.
- Ryan McDonald, Slav Petrov, and Keith Hall. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Edinburgh, Scotland, UK, July 2011. Association for Computational Linguistics. URL <https://aclanthology.org/D11-1006.pdf>.
- Ryan McDonald, Joakim Nivre, Yvonne Quirnbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 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 2013. Association for Computational Linguistics. URL <https://aclanthology.org/P13-2017.pdf>.
- Pruthwik Mishra, Vandan Mujadia, and Dipti Misra Sharma. POS tagging for resource poor languages through feature projection. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 50–55, Kolkata, India, December 2017. NLP Association of India. URL <https://aclanthology.org/W17-7507.pdf>.
- Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský, Amir Zeldes, and Daniel Zeman. CorefUD 1.0: Coreference meets Universal Dependencies. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid

- Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, H el ene Mazo, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4859–4872, Marseille, France, June 2022. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.520.pdf>.
- Jens Nilsson, Johan Hall, and Joakim Nivre. MAMBA meets TIGER: Reconstructing a Swedish treebank from antiquity. In *Proceedings of the NODALIDA Special Session on Treebanks*, 2005. URL <http://www.msi.vxu.se/users/nivre/research/Talbanken05.html>.
- Joakim Nivre and Chiao-Ting Fang. Universal Dependency evaluation. In Marie-Catherine de Marneffe, Joakim Nivre, and Sebastian Schuster, editors, *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 86–95, G otteborg, Sweden, May 2017. Association for Computational Linguistics. URL <https://aclanthology.org/W17-0411v2.pdf>.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Haji c, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 1659–1666, Portoro , Slovenia, May 2016. European Language Resources Association (ELRA). URL <https://aclanthology.org/L16-1262.pdf>.
- Joakim Nivre, Paola Marongiu, Filip Ginter, Jenna Kanerva, Simonetta Montemagni, Sebastian Schuster, and Maria Simi. Enhancing universal dependency treebanks: A case study. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 102–107, Bruxelles, Belgium, November 2018. Association for Computational Linguistics. URL <https://aclanthology.org/W18-6012.pdf>.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Haji c, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May 2020. European Language Resources Association (ELRA). URL <https://aclanthology.org/2020.lrec-1.497.pdf>.
- Atul Kr. Ojha and Daniel Zeman. Universal Dependency treebanks for low-resource Indian languages: The case of Bhojpuri. In *Proceedings of the WILDRE5– 5th Workshop on Indian Language Data: Resources and Evaluation*, pages 33–38, Marseille, France, May 2020. European Language Resources Association (ELRA). URL <https://aclanthology.org/2020.wildre-1.7.pdf>.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106,

2005. doi: 10.1162/0891201053630264. URL <https://aclanthology.org/J05-1004>.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, İstanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/274_Paper.pdf.
- Martin Popel, David Mareček, Jan Štěpánek, Daniel Zeman, and Zdeněk Žabokrtský. Coordination structures in dependency treebanks. In Hinrich Schuetze, Pascale Fung, and Massimo Poesio, editors, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 517–527, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL <https://aclanthology.org/P13-1051.pdf>.
- Martin Popel and Zdeněk Žabokrtský. TectoMT: Modular NLP framework. In *Advances in Natural Language Processing. 7th International Conference on NLP, IceTAL 2010, Reykjavik, Iceland, August 16-18, 2010, Proceedings*, pages 293–304, Reykjavík, Iceland, August 2010. Springer. doi: 10.1007/978-3-642-14770-8_33. URL https://ufal.mff.cuni.cz/~popel/papers/2010_icetal.pdf.
- Loganathan Ramasamy. *Parsing under-resourced languages: Cross-lingual transfer strategies for Indian languages*. PhD thesis, Univerzita Karlova v Praze, Praha, Czechia, 2014. URL <http://ufal.mff.cuni.cz/biblio/attachments/2014-ramasamy-m251064576937367469.pdf>.
- Rudolf Rosa. *Discovering the structure of natural language sentences by semi-supervised methods*. PhD thesis, Univerzita Karlova v Praze, Praha, Czechia, 2018. URL <http://ufal.mff.cuni.cz/biblio/attachments/2018-rosa-p4772924917445474076.pdf>.
- Rudolf Rosa, Jan Mašek, David Mareček, Martin Popel, Daniel Zeman, and Zdeněk Žabokrtský. HamleDT 2.0: Thirty dependency treebanks stanfordized. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2334–2341, Reykjavík, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/915_Paper.pdf.
- Rudolf Rosa, Daniel Zeman, David Mareček, and Zdeněk Žabokrtský. Slavic forest, norwegian wood. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 210–219, Valencia, Spain, April 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-1226. URL <https://aclanthology.org/W17-1226.pdf>.
- Sebastian Schuster and Christopher D. Manning. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth*

- International Conference on Language Resources and Evaluation (LREC'16)*, pages 2371–2378, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL <https://aclanthology.org/L16-1376.pdf>.
- Abishek Stephen and Daniel Zeman. Universal Dependencies for Malayalam. *The Prague Bulletin of Mathematical Linguistics*, (120):31–46, 2023. ISSN 0032-6585. URL <https://ufal.mff.cuni.cz/pbml/120/art-stephen-zeman.pdf>.
- John Sylak-Glassman, Christo Kirov, David Yarowsky, and Roger Que. A language-independent feature schema for inflectional morphology. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 674–680, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2111. URL <https://aclanthology.org/P15-2111.pdf>.
- Dima Taji, Nizar Habash, and Daniel Zeman. Universal Dependencies for Arabic. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 166–176, Valencia, Spain, April 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-1320. URL <https://aclanthology.org/W17-1320.pdf>.
- Mariona Taulé, M. Antònia Martí, and Marta Recasens. AnCora: Multilevel annotated corpora for Catalan and Spanish. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/35_paper.pdf.
- Ulf Teleman. Manual för grammatisk beskrivning av talad och skriven svenska (Mamba), 1974.
- Lucien Tesnière. *Éléments de syntaxe structurale*. Klincksieck, Paris, France, 1959.
- Jörg Tiedemann. Rediscovering annotation projection for cross-lingual parser induction. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1854–1864, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL <https://aclanthology.org/C14-1175.pdf>.
- Marsida Toska, Joakim Nivre, and Daniel Zeman. Universal Dependencies for Albanian. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 178–188, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics. URL <https://aclanthology.org/2020.udw-1.20.pdf>.

- Francis Tyers and Karina Mishchenkova. Dependency annotation of noun incorporation in polysynthetic languages. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 195–204, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics. URL <https://aclanthology.org/2020.udw-1.22.pdf>.
- Jens E. L. Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O’Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. Designing a Uniform Meaning Representation for natural language processing. *Künstliche Intelligenz*, 35(0):343–360, October 2021. ISSN 1610-1987. doi: 10.1007/s13218-021-00722-w. URL <https://par.nsf.gov/servlets/purl/10288899>.
- David Yarowsky and Grace Ngai. Inducing multilingual pos taggers and np brackets via robust projection across aligned corpora. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, PA, USA, June 2001. Association for Computational Linguistics. URL <https://aclanthology.org/N01-1026.pdf>.
- Zdeněk Žabokrtský, Daniel Zeman, and Magda Ševčíková. Sentence meaning representations across languages: What can we learn from existing frameworks? *Computational Linguistics*, 46(3):605–665, September 2020. doi: 10.1162/colia_00385. URL <https://aclanthology.org/2020.cl-3.3.pdf>.
- Zdeněk Žabokrtský, Niyati Bafna, Jan Bodnár, Lukáš Kyjánek, Emil Svoboda, Magda Ševčíková, and Jonáš Vidra. Towards universal segmentations: UniSegments 1.0. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1137–1149, Marseille, France, June 2022. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.122.pdf>.
- Daniel Zeman. *Parsing with a Statistical Dependency Model*. PhD thesis, Univerzita Karlova v Praze, Praha, Czechia, 2004. URL <http://ufal.mff.cuni.cz/biblio/attachments/2004-zeman-m5440617933930313730.pdf>.
- Daniel Zeman. Reusable tagset conversion using tagset drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*, pages 213–218, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/66_paper.pdf.
- Daniel Zeman. Slovak dependency treebank in Universal Dependencies. *Journal of Linguistics / Jazykovedný časopis*, 68(2):385–395, December 2017. doi: 10.1515/jazcas-2017-0048. URL <https://sciendo.com/article/10.1515/jazcas-2017-0048>.

- Daniel Zeman. *The World of Tokens, Tags and Trees*. ÚFAL MFF UK, Praha, Czechia, 2018. ISBN 978-80-88132-09-7. URL https://ufal.mff.cuni.cz/books/preview/2018-zeman_full.pdf.
- Daniel Zeman. Subword relations, superword features. In *UniDive General Meeting at Paris-Saclay posters*, Orsay, France, February 2023. URL https://unidive.lisn.upsaclay.fr/lib/exe/fetch.php?media=meetings:2023-saclay:abstracts:39_zeman_subword_relations_superword_features.pdf. <https://unidive.lisn.upsaclay.fr/lib/exe/fetch.php?media=meetings:2023-saclay:abstracts:wg1-2-zeman-poster.pdf>.
- Daniel Zeman and Philip Resnik. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42, Hyderabad, India, January 2008. Asian Federation of Natural Language Processing. URL <https://aclanthology.org/I08-3008.pdf>.
- Daniel Zeman, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zdeněk Žabokrtský, and Jan Hajič. HamleDT: To parse or not to parse? In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2735–2741, İstanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/429_Paper.pdf.
- Daniel Zeman, Ondřej Dušek, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zdeněk Žabokrtský, and Jan Hajič. HamleDT: Harmonized multi-language dependency treebank. *Language Resources and Evaluation*, 48:601–637, 2014. doi: 10.1007/s10579-014-9275-2. URL <https://link.springer.com/content/pdf/10.1007/s10579-014-9275-2.pdf>.
- Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Drohanova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-3001. URL <https://aclanthology.org/K17-3001v1.pdf>.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 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, Bruxelles, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-2001. URL <https://aclanthology.org/K18-2001v1.pdf>.

List of Tables

1	Morphological / POS tag examples for various languages. The tags for adjectives as defined in the Penn Treebank [Marcus et al., 1993], Mamba [Teleman, 1974, Nilsson et al., 2005], Stockholm-Umeå Corpus [Gustafson-Capková and Hartmann, 2006, p. 20–21], and the Prague Dependency Treebank (PDT) [Hajič et al., 2000]. The three PDT tags represent only a fraction; as many as 378 feature combinations are possible in a regular adjective paradigm. Stockholm-Umeå is less rich, but still it has many more tags than the three displayed here.	3
2.1	Interset features and their values.	13
2.2	The nominative and genitive forms of Croatian 3rd person pronouns, and the nominative forms of the corresponding possessive pronouns.	18