

Universal Dependencies are neither Universal nor Dependencies

Joakim Nivre

Uppsala University

Department of Linguistics and Philology

joakim.nivre@lingfil.uu.se

Abstract

Universal Dependencies is a framework for cross-linguistically consistent treebank annotation, using binary relations to represent syntactic structure. Despite the name of the framework, it should not be assumed that the relation types are all universal, nor that they are all asymmetric dependency relations in a narrow sense. The purpose of this paper is to explain why and to clear up a few common misconceptions about Universal Dependencies.

1 Introduction

Universal Dependencies (UD) is a framework for morphosyntactic annotation, designed to be applicable to all human languages and to enable meaningful cross-linguistic comparisons. The two versions of the guidelines are described in Nivre et al. (2016) and Nivre et al. (2020); a longer description of the underlying linguistic theory can be found in De Marneffe et al. (2021); and annotated data for 168 languages¹ can be found together with additional documentation on the UD website.²

One of the basic design principles of UD is that syntactic structure can be represented by trees that are composed of binary syntactic relations. By way of example, Figure 1 shows a UD representation for the English sentence *Sherlock Holmes will solve the mystery in spite of the strange clues*. The syntactic representation is a tree rooted at the word *solve*, and all edges of the tree are labeled with syntactic relation types such as nominal subject (*nsubj*), direct object (*obj*), and so on. In addition, each word is assigned a part-of-speech tag like NOUN, VERB, ADJ, etc.³

¹UD v2.15, released November 15, 2024.

²<https://universaldependencies.org>

³In the full UD representation, each word is also assigned

The syntactic relations used in UD often correspond to the asymmetric head-dependent relations that are central to the dependency grammar tradition, but UD does not assume that all syntactic relations are dependency relations in this sense. This has led to some criticism of the UD framework from proponents of dependency grammar, based on the observation that UD relations sometimes appear to violate commonly held assumptions among dependency grammarians (see, for example, Gerdes and Kahane, 2016; Gerdes et al., 2018; Osborne and Gerdes, 2019). The first point of this paper is to clarify the status of syntactic relations in UD in this respect.

UD is designed to be applicable to all natural languages, and the same should hold of the inventory of syntactic relation types posited in UD. However, this does not mean that all relations are assumed to be universal in the sense that they can be found in all languages. Clarifying this issue and relating it to research in linguistic typology is the second point of this paper. Putting the two points together gives us the apparent contradiction in the title of the paper, which we will now try to resolve.

Before proceeding, however, I want to clarify that the views expressed in this paper are my own and sometimes go beyond the official UD documentation. I have arrived at these views by reflecting on the evolution of UD, as well as on the criticism that it has received, and this has helped me deepen my understanding of UD. My hope is that readers may find it somewhat helpful as well.

2 UD Relations are not Dependencies

Dependency grammar is a family of syntactic theories and frameworks, which all assume that an important part of natural language syntax can be analyzed in terms of binary asymmetrical relations between a *head* and a *dependent*. Such relations are typically represented as a *head* and a *dependent* lemma and a set of morphological features, which have been suppressed in Figure 1 for readability.

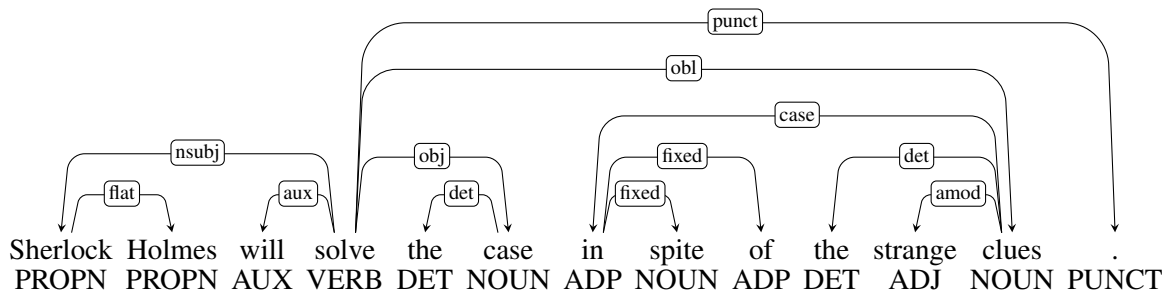


Figure 1: Simplified UD representation of an English sentence.

lations are called *dependency* relations, because of the asymmetric relation, where the dependent in some way presupposes the head, but not vice versa. However, it has been hard to reach consensus on the criteria for distinguishing dependency relations, and for identifying the head and dependent in different constructions (see, for example, Nivre, 2006, chapter 3).

The basic idea underlying dependency grammar has been traced back to the Indian grammarian Pāṇini several centuries before the common era, and modern elaborations of this idea can be found in Tesnière (1959), Hudson (1984), Sgall et al. (1986), and Mel’čuk (1988), among others. Theories in this family all agree on the central role of dependency relations, but differ in many of the details, and not all of them assume that syntactic structure can be analyzed only in terms of dependency relations. UD follows Tesnière (1959) and Hudson (1984) in *not* assuming that all syntactic relations are dependency relations.

Of the 37 syntactic relations in the UD inventory,⁴ the ones that clearly fulfill the criteria of dependency relations are argument and modifier relations, which are also the relations which all varieties of dependency grammar tend to agree on. Thus, in Figure 1, the subject (*nsubj*) and object (*obj*) relations are dependency relations that hold between the head verb *solve* and the dependent arguments *Sherlock Holmes* and *the case*, respectively. Similarly, the oblique modifier (*obl*) relation relates the head verb *solve* to the dependent *in spite of the confusing clues*, and the adjectival modifier (*amod*) relates the head noun *clues* to the dependent *strange*. But the remaining relations are not straightforward dependency relations.

Perhaps the most obvious example of a non-dependency relation in UD is the relation called

flat, which, as the name suggests, is used to connect two elements of a sentence that belong together but where neither of them can be identified as the head. Hence, it denotes a *symmetric* relation between two elements, not an asymmetric dependency relation. An instance of the *flat* relation can be found in Figure 1, where the name *Sherlock Holmes* is analyzed as a headless phrase, because standard tests for dependency does not clearly identify either the first or the last name as the syntactic head of the phrase.⁵ Another clear non-dependency relation is the *fixed* relation, which combines words that were historically connected by syntactic relations but which have frozen into a fixed expression, such as *in spite of* in Figure 1, which functions as a preposition introducing an oblique modifier despite on the surface being a preposition-noun-preposition sequence.

Slightly more controversial are the relations holding between content words and function words, such as the auxiliary (*aux*) relation from *solve* to *will*, the determiner (*det*) relation from *case* to *the* (and from *solutions* to *the*), and the case marker (*case*) relation from *solutions* to *in spite of*. UD follows Tesnière (1959) in assuming that such relations are not really dependency relations,⁶ but instead are relations that allow multiple words to form syntactic nuclei in a way analogous to morphological inflection of content words. Thus, the function words involved in *aux*, *det* and *case* relations often correspond to morphological affixes in other languages, or even in the same language.

The UD treatment of function words has been the subject of some debate (see, for example, Gerdes and Kahane, 2016; Gerdes et al., 2018;

⁵One such test is substitutability of the head for the whole phrase, and in this case both *Sherlock* and *Holmes* can replace the whole phrase without loss of grammaticality or drastic change of meaning.

⁶For example, they fail the usual substitution test in that none of the elements can replace the whole.

⁴<https://universaldependencies.org/u/dep/index.html>

Osborne and Gerdes, 2019), because it appears to violate the assumption that function words are syntactic heads, which is found in many versions of dependency grammar (as well as syntactic theories in other traditions). In my view, this criticism rests on a misconception, because UD does not assume that function words are dependents of content words, nor the other way around. Instead, content words together with their associated function words form the syntactic units that enter into dependency relations (and other syntactic relations).⁷

To be fair, it is not hard to see how these misconceptions may arise. Besides the name of the framework, the nature of the tree-shaped syntactic representations,⁸ where every word except one is attached to another word with a directed edge, may give the impression that every word except one is a syntactic dependent of another word. In addition, the UD guidelines have not always been clear about the status of syntactic relations, often using the terms *head* and *dependent* as convenient shorthands for *parent node* and *child node* in the syntactic tree. These tree-shaped representations are a heritage from the parsing community from which UD originates and are better described as *spanning trees*, a term used in the graph-based approach to dependency parsing (McDonald et al., 2005a,b; McDonald and Pereira, 2006). This term only implies that every word of a sentence is included in the tree, but remains neutral about the nature of the pairwise syntactic relations. These relations are instead specified by syntactic relation types, encoded in labels like *nsubj*, *obj*, *flat*, and *fixed*. And while some of these relation types fulfill the criteria of dependency relations, others clearly do not. Hence, UD relations are not (all) dependencies.

3 UD Relations are not Universal

Syntactic relation types in UD should be cross-linguistically applicable, meaning that it should be possible to identify instances of them without relying on language-specific criteria. However, this does not mean that the relations are universal in

⁷An elaboration of the UD position on function words can be found in De Marneffe et al. (2024). An overview of the treatment of function words in different versions of dependency grammar can be found in Osborne (ed.) (2024).

⁸The tree constraint holds for *basic* UD representations; there is also an *enhanced* representation, which takes the form of a general graph, which will not be discussed here.

the stronger sense that they are assumed to exist in all languages – at least not all of them.

To explain this distinction, it may be helpful to introduce some concepts from linguistic typology. Croft (2022) distinguishes two types of comparative concepts, which can be used when comparing languages typologically:

construction: any pairing of form and function in a language (or any language) used to express a particular combination of semantic content and information packaging

strategy: a construction in a language (or any language), used to express a particular combination of semantic content and information packaging (the ‘what’), that is further distinguished by certain characteristics of grammatical form that can be defined in a crosslinguistically consistent fashion (the ‘how’)

A construction, in Croft’s sense, is universal and defined only in terms of its function (which in turn is defined in terms of semantic content and information packaging), while a strategy is a particular way of realizing this function morphosyntactically, and hence not universal.

To exemplify these concepts, let us consider the *predicate nominal* construction, which is “a clause construction defined by the function of predicating an object concept of a referent – that is, asserting what object category the referent belongs to”.⁹ Two common strategies for this construction are exemplified in (1) and (2–3).

- (1) ИВАН ТАНЦОР
Ivan.NOM dancer.NOM
‘Ivan is a dancer’
- (2) Ivan är dansare
Ivan COP dancer
‘Ivan is a dancer’
- (3) Ivan is a dancer
Ivan COP a dancer

The Russian example in (1) uses a *zero* strategy (Stassen, 1997), which simply juxtaposes the referring expression ИВАН with the noun ТАНЦОР in nominative case expressing the object concept. By contrast, the Swedish and English examples in (2) and (3) both use a *verbal copula* strategy (Stassen, 1997), where predication is mediated by a copula verb. The notion of strategy allows us to abstract

⁹<https://comparative-concepts.github.io/cc-database/>

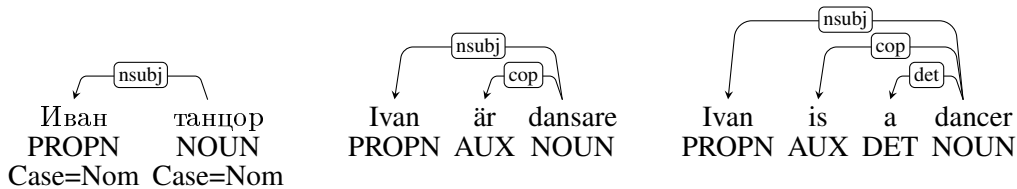


Figure 2: Simplified UD annotation for predicate nominal constructions in Russian, Swedish and English.

over language-specific constructions and say that Swedish and English use the same strategy, while the Russian strategy is different.

UD needs to annotate elements of both constructions and strategies, but since it is a framework for morphosyntactic annotation, it must often give priority to strategies. Figure 2 shows how UD can represent the predicate nominal examples in Russian, Swedish, and English. What is common to all three languages is the subject (*nsubj*) relation from the nominal predicate to the referent, which can therefore be said to capture the core of the predicate nominal construction (even though it is also found as a component of other constructions). By contrast, the copula (*cop*) relation is found only in Swedish and English, and is clearly an element of a non-universal strategy.

By and large, argument and modifier relations tend to be part of constructions and are therefore universal. For example, all languages have intransitive and transitive clauses, the arguments of which instantiate subject (*nsubj*) and object (*obj*) relations, although the way that these relations map on to morphosyntactic markers varies across languages, with nominative-accusative and ergative-absolutive alignment as the main strategies. Similarly, all languages can express property modification and object modification, involving the adjectival (*amod*) and nominal (*nmod*) relations, respectively, although the morphosyntactic encoding by means of case markers or indexation again varies depending on strategies.

At the other end of the spectrum, we find relations that denote specific strategies, or strategy elements, and which are therefore not universal. These include the copula (*cop*) relation discussed above, as well as several of the function word relations discussed in Section 2. Thus, the case marker (*case*) relation is used for case markers (including adpositions) that are realized as free morphemes, a strategy that is not universal. Similarly, not all languages encode tense, aspect, mood, or evidentiality using independent words, which is what the

auxiliary (*aux*) relations is used for. Hence, UD relations are not (all) universal.

4 Conclusion

In this paper, I have tried to explain why it is a mistake to assume that all syntactic relations in UD are dependency relations in the narrow sense and that they are universal in a strong sense. It follows that *Universal Dependencies* must be treated as a proper name, not as a descriptive phrase, which is why the title of this paper is not a contradiction.

Acknowledgments

The work presented in this paper was supported by Swedish Research Council grant no. 2022-02909. I am grateful to the editor for comments on the first draft of the paper.

References

- William Croft. 2022. *Morphosyntax: Constructions of the World’s Languages*. Cambridge University Press.
- Marie-Catherine De Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47:255–308.
- Marie-Catherine De Marneffe, Joakim Nivre, and Daniel Zeman. 2024. Function words in Universal Dependencies. *Linguistic Analysis*, 43(3–4):549–588.
- Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2018. SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 66–74.
- Kim Gerdes 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.
- Richard A. Hudson. 1984. *Word Grammar*. Blackwell.

- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 91–98.
- Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 81–88.
- Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 523–530.
- Igor Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press.
- Joakim Nivre. 2006. *Inductive Dependency Parsing*. Springer.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Dan Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, pages 1659–1666.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Dan Zeman. 2020. Universal Dependencies v2: An ever-growing multilingual treebank collection. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC)*, pages 4034–4043.
- Timothy Osborne and Kim Gerdes. 2019. The status of function words in dependency grammar: A critique of Universal Dependencies (UD). *Glossa*, 4(1):17.
- Timothy Osborne (ed.). 2024. The status of function words in dependency grammar. *Linguistic Analysis*, 43(3–4).
- Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Pragmatic Aspects*. Reidel.
- Leon Stassen. 1997. *Intransitive Predication*. Oxford University Press.
- Lucien Tesnière. 1959. *Éléments de syntaxe structurale*. Editions Klincksieck.