ANNUAL REVIEWS

*Annual Review of Linguistics*

# Dependency Grammar

Marie-Catherine de Marneffe[1] and Joakim Nivre[2]

[1]Department of Linguistics, The Ohio State University, Columbus, Ohio 43210, USA; email: demarneffe.1@osu.edu

[2]Department of Linguistics and Philology, Uppsala University, 75267 Uppsala, Sweden; email: joakim.nivre@lingfil.uu.se

Annu. Rev. Linguist. 2019. 5:197–218

The *Annual Review of Linguistics* is online at linguist.annualreviews.org

https://doi.org/10.1146/annurev-linguistics-011718-011842



## Keywords

dependency grammar, dependency frameworks, dependency parsing, Universal Dependencies

## Abstract

Dependency grammar is a descriptive and theoretical tradition in linguistics that can be traced back to antiquity. It has long been influential in the European linguistics tradition and has more recently become a mainstream approach to representing syntactic and semantic structure in natural language processing. In this review, we introduce the basic theoretical assumptions of dependency grammar and review some key aspects in which different dependency frameworks agree or disagree. We also discuss advantages and disadvantages of dependency representations and introduce Universal Dependencies, a framework for multilingual dependency-based morphosyntactic annotation that has been applied to more than 60 languages.

## 1. INTRODUCTION

Dependency grammar is probably best described as a particular perspective on linguistic analysis, in particular syntactic analysis, rather than as a coherent theoretical framework. This perspective can be found in many different traditions in the history of linguistics, going back as far as Pāṇini's grammar of Sanskrit several centuries before the Common Era (Kruijff 2002), and also reflected in certain medieval theories of grammar (Covington 1984). During the twentieth century, dependency grammar developed largely as a form of syntactic representation used by traditional grammarians, especially in Europe, and particularly in classical and Slavic linguistics (Mel'čuk 1988). One of the earliest attempts to build a complete theory of grammar based on dependency is the seminal book by Tesnière (1959), published posthumously, which also demonstrates the potential of dependency grammar to capture similarities and differences across languages, a theme we return to below. Tesnière's research is usually taken as the starting point of the modern theoretical tradition of dependency grammar.

The common core of all varieties of dependency grammar is the assumption that syntactic structure consists primarily of binary asymmetrical relations, called dependency relations (or dependencies, for short), that hold between words. This structure can be displayed in a dependency tree, where nodes represent words and labeled arcs represent different types of dependency relations (**Figure 1**).[1]

A dependency tree representation of syntactic structure emphasizes the functional role of a word in a sentence. Thus, the noun *dogs* in **Figure 1** fills the subject role of the verb *chase*, while the noun *cats* fills the object role. The adjective *small* is an attributive modifier of *dogs*, while the adverb *happily* is an adverbial modifier of *chase*.[2] In all cases, however, the relation is asymmetric, with one word, called the dependent, complementing or modifying the other, called the head.[3] Thus, dependency trees prioritize functional dependency relations over structural constituency relations, and words over phrases, which distinguishes them from other types of syntactic representations, in particular phrase-structure or constituency trees (Matthews 1981).

The idea that syntactic structure can be reduced to binary dependency relations is what unites the dependency grammar tradition. However, over time this idea has been refined in several different ways as different theoretical frameworks have developed, and today there are a number of issues on which different theoretical approaches to dependency grammar do not agree. We use these issues to organize the discussion of different dependency grammar frameworks in Section 2. This allows us to cover some of the most central issues in dependency grammar and to present examples of the most influential frameworks, but without providing a comprehensive survey of all frameworks, which is outside the scope of this review.

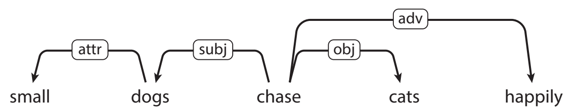


**Figure 1**
A dependency tree for the sentence *small dogs chase cats happily*.

[1] There are several different conventions for drawing dependency trees. We adopt the one that is most popular in computational linguistics, with dependency arcs drawn in the half-plane above the sequence of words and arcs pointing away from the root.

[2] The labels used in examples, unless stated otherwise, refer to traditional grammatical functions and are not taken from any specific dependency grammar framework.

[3] Other terms currently in use are governor (instead of head) and modifier (instead of dependent).

After comparing different dependency grammar frameworks in Section 2, in Section 3 we turn to the advantages offered by the dependency grammar perspective for linguistic analysis and natural language processing (NLP). We analyze the reasons for the steadily increasing interest in dependency grammar during the last few decades, especially in the NLP community, and discuss some of the limitations of the approach.

Finally, we devote Section 4 to Universal Dependencies (UD), a recent initiative to create crosslinguistically consistent morphosyntactic annotation for many languages, based largely on the dependency grammar perspective. Although UD began as an initiative in the NLP community, it has gradually attracted more and more interest from linguistic researchers, especially those in linguistic typology.

## 2. THEORETICAL ISSUES AND FRAMEWORKS IN DEPENDENCY GRAMMAR

Before we begin our discussion of theoretical issues and frameworks, a clarification about the term grammar seems in order. In formal and generative linguistics, this term is often understood in the sense of a formal declarative system capable of generating languages and making predictions about grammaticality. Most frameworks in the dependency grammar tradition do not make use of grammars in this sense but are better described as syntactic analysis schemes that may be more or less formalized. These characteristics carry over to most computational models for dependency parsing, which rely on machine learning techniques to learn statistical regularities from annotated corpora without inducing a formal grammar (Kübler et al. 2009).[4]

Keeping in mind that dependency grammar theories are not always fully formalized, we now consider some of the issues that separate different dependency grammar frameworks:

- Is the notion of dependency sufficient as well as necessary for syntactic analysis?
- How many levels of dependency structure do we need to distinguish?
- What is the nature of lexical elements and dependency relations?
- What are the formal properties of dependency representations?
- What are the criteria for determining dependency relations?

### 2.1. Sufficiency of Dependency

The first and perhaps most fundamental question is whether all of syntactic structure can be reduced to dependency relations. According to Tesnière (1959, 2015), the answer is clearly no. Tesnière distinguishes two relations, aside from dependency, that can hold between words in a sentence: junction (French *jonction*) and transfer (*translation*). Junction is the relation that holds between coordinated items that are dependents of the same head or heads of the same dependent, and transfer is the relation that holds between a lexical element and a function word or other element that changes the syntactic category of the lexical element so that it can enter into different dependency relations. An example of the latter is the relation that holds between the preposition *de* and *Pierre* in the construction *le livre de Pierre* ('Pierre's book,' literally 'the book of Pierre'), where the preposition *de* allows the proper name *Pierre* to modify a noun—a dependency relation otherwise reserved for adjectives.

Another way in which theories may depart from a pure dependency analysis is in allowing a restricted form of constituency analysis, so that dependencies can hold between strings of words

[4]This is different from the constituency parsing tradition, where even data-driven approaches normally induce a formal grammar such as a probabilistic context-free grammar from an annotated corpus.

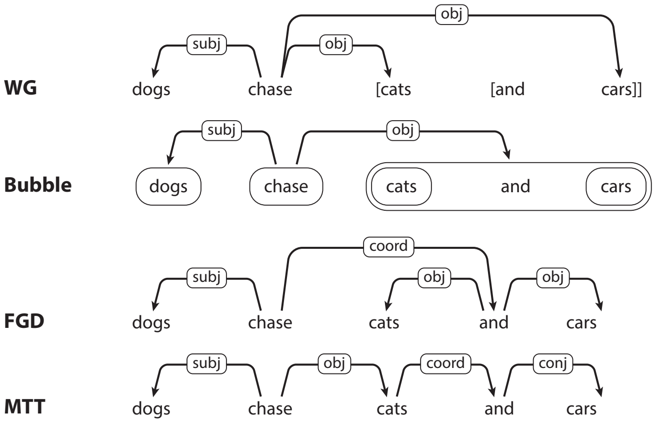


**Figure 2**

Treatment of coordination in four dependency frameworks: Word Grammar (WG), bubble tree (Bubble), Functional Generative Description (FGD), and Meaning–Text Theory (MTT).

rather than single words. This possibility is exploited, to different degrees, in Meaning–Text Theory (MTT) (Mel'čuk 1988, Milicevic 2006) and Word Grammar (WG) (Hudson 1984, 1990, 2007), notably in connection with coordination, as illustrated with a WG analysis in **Figure 2**, where the phrase *cats and cars* is bracketed into a "word string" that licenses the double-object dependency. An alternative way to avoid reducing coordination to dependency relations involves the use of a bubble tree, proposed by Kahane (1997), in which dependency relations may hold between sets of nodes (called bubbles), also shown in **Figure 2**.

Other frameworks, such as Functional Generative Description (FGD) (Sgall et al. 1986), instead assume that all constructions can be analyzed in terms of dependencies and introduce a special dependency relation, *coord*, that allows the coordinating conjunction to act as the syntactic head of the coordinate structure, as shown in **Figure 2**. MTT also assumes a dependency-based analysis of coordination, but treats the first conjunct as the head (see **Figure 2**), and admits bracketing when needed to disambiguate the scope of modifiers, as in phrases like *young men and women*, where the attribute *young* may modify either the first conjunct (*men*) or the whole coordination (*men and women*).

Computational models for dependency parsing are typically restricted to plain dependency trees without additional relation types or phrases (Kübler et al. 2009). In this case, however, the motivation comes more from a desire to simplify computational processing than from theoretical considerations, as discussed in Sections 3 and 4, below.

## 2.2. Levels of Representation

Another dividing line in dependency grammar is that between monostratal and multistratal theories, that is, between theories that posit a single level of representation and those that assume multiple levels, or strata. MTT and FGD are typical multistratal theories, the former positing at least five levels of representation, namely semantics (SemR), deep syntax (DSyntR), surface syntax (SSyntR), deep morphology (DMorphR), and surface morphology (SMorphR), and the latter at least three, specifically morphological, analytical (surface syntax), and tectogrammatical (deep syntax). Dependency structure is recognized at several levels, and the levels are related to one another by rules.

This is in contrast to WG, for example, which is a monostratal theory, and to computational parsing models, which can typically handle only a single dependency tree. For simplicity, we limit the discussion throughout this review to syntactic dependency representations, which we assume either are monostratal or make a binary distinction between surface and deep syntax. This means that we ignore both morphological and semantic dependency representations.

## 2.3. Lexical Elements and Relations

Another set of issues concerns the nature of the lexical elements and relations that combine into (syntactic) dependency structures. Or, more concretely, what exactly can be represented by a node and a labeled arc in a dependency tree, respectively? So far, we have made the simplifying assumption that nodes represent words and arcs represent traditional grammatical functions like subject and object, but this is only one of many possibilities.

To a certain extent, the choices here are dependent on other theoretical choices. For example, whereas theories that reduce all syntactic structures to dependencies tend to assume that a node corresponds to a single word, Tesnière (1959) assumes a more abstract notion of nucleus, defined as a set of words that together form a syntactic and semantic unit. A nucleus always contains a content word, which is the semantic core, and may also contain one or more function words that serve as transfer elements, in which case the nucleus is said to be dissociated. In addition, multistratal theories often posit more abstract elements like lexemes in deep syntax and concrete word forms in surface syntax.

A similar variation is found for dependency relations, where surface syntactic representations tend to use traditional grammatical functions (e.g., subject, object, adverbial), while deeper levels employ more semantically oriented role types (e.g., agent, patient, goal) belonging to the tradition of case roles or thematic roles (Fillmore 1968, Jackendoff 1972, Dowty 1989). An alternative scheme of representation, found in MTT but also in Tesnière's research, involves the use of numerical indices for valency-bound dependents to reflect a canonical ordering of arguments (argument 1, 2, 3, etc.) and the use of descriptive labels only for valency-free dependents.

Label sets differ greatly in their granularity. For instance, in the Prague Dependency Treebank scheme, which is based on FGD, the surface syntactic (analytical) layer uses only 17 distinct labels, while the deep syntactic (tectogrammatical) layer has more than 50 different labels to capture deep syntactic and semantic relations. UD (Nivre et al. 2016), a framework specifically designed for multilingual annotation, uses 37 relation labels but allows subtypes to capture more fine-grained language-specific constructions.

Finally, it is possible to have unlabeled dependency trees, that is, dependency trees without any labels at all. Unlabeled trees are sometimes used as intermediary or underspecified representations, especially in NLP, but they are not very informative from a linguistic point of view since they lack information about the precise functional relations between heads and dependents. Nevertheless, when evaluating dependency parsers, it is customary to report both labeled and unlabeled accuracy (Nivre et al. 2007).

## 2.4. Formal Properties of Dependency Representations

So far, we have assumed that a (syntactic) dependency representation is a tree, but this is by no means always assumed. For example, WG assumes a more general graph structure, where a node may have multiple incoming arcs and where the graph may even be cyclic. In MTT and FGD, both deep and surface syntactic representations are trees, but only the latter has a one-to-one correspondence between nodes and words. In the NLP community, most research on dependency parsing has assumed that the output should be a tree spanning all the words of the sentence (Yamada & Matsumoto 2003, Nivre 2003, McDonald et al. 2005b). The notion of spanning tree implies that every word is included as a node in the tree (the spanning condition) and that every node except the root has exactly one incoming arc (the tree condition). However, there seems to be increasing interest in parsing to more complex graph structures, especially to obtain general-purpose semantic representations in the shape of dependency graphs (Oepen et al. 2014, 2015).

Even if we limit our attention to dependency trees, a number of open issues remain, in particular concerning the relation between dependency structure and word order. According to Tesnière (1959), dependency relations belong to the structural order, which differs from the linear order of a spoken or written string of words. Most versions of dependency grammar follow Tesnière in assuming that the nodes of a dependency structure are not linearly ordered in themselves but only in relation to a particular surface realization of this structure. A notable exception to this generalization is FGD, in which the representations of both the analytical layer and the tectogrammatical layer are linearly ordered to capture aspects of information structure (Sgall et al. 1986).

However, whether or not dependency relations introduce a linear ordering, there may be constraints relating dependency structures to linear realizations. The best-known example is the constraint of projectivity, discussed early on by Lecerf (1960), Hays (1964), Gaifman (1965), and Marcus (1965). A dependency tree is projective with respect to a particular linear order of the nodes if, for every arc $h \rightarrow d$ and node $w$, $w$ occurs between $h$ and $d$ in the linear order only if $w$ is dominated by $h$ (where dominates is the reflexive and transitive closure of the arc relation). Alternatively, we can say that a dependency tree is projective if the yield of every subtree forms a contiguous substring of the linear order (where the yield is, again, the reflexive and transitive closure of the arc relation). The tree in **Figure 3** is nonprojective because the word *dogs* appears between the head *bigger* and the dependent *mine* but is not dominated by *bigger*. Equivalently, the yield of the subtree rooted at *bigger* is *bigger than mine*, which is not a contiguous substring.

Broadly speaking, most dependency grammar theories regard projectivity as the norm but also recognize the need for nonprojective trees in order to capture certain phenomena, for example, long-distance dependencies (Mel'čuk 1988, Hudson 1990). Some multistratal theories allow non-projective trees in some layers but not in others. For instance, FGD assumes that tectogrammatical representations are projective while analytical representations need not be (Sgall et al. 1986).

When it comes to computational parsing models, assuming projectivity is advantageous from the point of view of runtime complexity. With this constraint, standard dynamic programming algorithms for bottom-up parsing run in polynomial time (Eisner 1996, McDonald et al. 2005a, Koo & Collins 2010), and greedy shift–reduce algorithms in linear time (Nivre 2003). Without this constraint, there are no polynomial-time dynamic programming algorithms, and even greedy parsers have quadratic complexity in the worst case. Nevertheless, several models have been proposed to handle nonprojective trees efficiently using various approximation techniques (McDonald et al. 2005b, Nivre & Nilsson 2005, Attardi 2006, Martins et al. 2009, Nivre 2009, Koo et al. 2010).

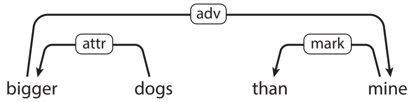


**Figure 3**

Nonprojective dependency tree for the phrase *bigger dogs than mine*.

Recent research in computational linguistics has also been concerned with finding subclasses of nonprojective trees that strike a good balance between computational complexity and linguistic coverage, such as well-nested trees with bounded block degree (Kuhlmann & Möhl 2007), 2-planar trees (Gómez-Rodríguez & Nivre 2010), gap minding trees (Pitler 2012) and 1-endpoint-crossing trees (Pitler et al. 2013). Some of these classes have clear linguistic motivation. For example, well-nested trees with bounded block degree correspond to mildly context-sensitive grammar formalisms like Tree-Adjoining Grammar (Joshi 1985), and Pitler et al. (2013) conjecture that the 1-endpoint-crossing property is implied by the phrase-impenetrability condition (Chomsky 1998).

## 2.5. Criteria for Dependency Relations

The final issue we consider concerns two central questions: What are the criteria for establishing that a dependency relation exists between two words or other lexical elements? And how do we determine which word is the head and which is the dependent? Let us begin by noting that these questions are relevant not only for dependency grammar but for any approach to syntax that recognizes a notion of syntactic head, including all constituency-based frameworks that subscribe to some version of $\overline{\text{X}}$ theory (Chomsky 1970, Jackendoff 1977), such as Head-Driven Phrase-Structure Grammar (Pollard & Sag 1987, 1994). Here are some of the criteria that have been proposed for identifying a syntactic relation between a head H and a dependent D in a construction C (Zwicky 1985, Hudson 1990):

- H determines the syntactic category of C and can often replace C.
- H determines the semantic category of C; D provides semantic specification.
- H is obligatory; D may be optional.
- H selects D and determines whether D is obligatory or optional.
- The form of D depends on H (agreement or government).
- The linear position of D is specified with reference to H.

It is clear that this list contains a mix of different criteria, some syntactic and some semantic, and one may ask whether there is a single coherent notion of dependency corresponding to all the different criteria. This has led some theorists, such as Hudson (1990), to suggest that the concept of head has a prototype structure, where typical instances satisfy all or most of the criteria while more peripheral instances satisfy fewer. Other authors have instead emphasized the need to distinguish different kinds of dependency relations. According to Mel'čuk (1988), for example, the words of a sentence can be linked by three types of dependencies—morphological, syntactic, and semantic—which also motivates the need for multiple levels of representation. Even if we limit our attention to syntactic dependencies, there may be different criteria for deep syntax, which is often assumed to be universal, and surface syntax, which varies across languages.

Without going into a detailed discussion of criteria for dependency relations, we can conclude that there is a core of syntactic constructions on which all dependency grammar theories agree. On the one hand, this core includes exocentric constructions involving a predicate, in particular a verb, and its arguments. These constructions are exocentric in the sense that none of the constituents can replace the whole, but all dependency grammars assume that the predicate is a head that licenses its arguments through the notion of valency. Thus, in **Figure 1**, the subject *dogs* and the object *cats* are uncontroversially dependents of the verb *chase*. On the other hand, this core also includes endocentric constructions, where one element is syntactically equivalent to the whole (and can replace the whole without loss of grammaticality). Thus, the adjectival modifier *small* and the adverbial modifier *happily* are universally assumed to be dependents of the noun *dogs* and the verb *chase*, respectively.

When we move outside the core of uncontroversial constructions, the different positions adopted by different frameworks are often related to issues that are discussed above. For example, while a multistratal theory can posit different dependency structures at different levels, a monostratal theory has to prioritize some criteria over others. It is also striking that some of the most intense discussions concern phenomena where there is not even universal agreement that they can be analyzed in terms of dependencies at all. Let us conclude this section by briefly discussing two such phenomena: coordination and function words.

All natural languages appear to have a way of forming phrases by coordinating two similar phrases, with or without an overt coordinator, but the analysis of such phrases remains controversial in dependency grammar. For Tesnière (1959) and Hudson (1990), coordination simply falls outside the notion of dependency. By contrast, both FGD and MTT treat coordination as a dependency structure but disagree about which element is the head. In FGD, it is the coordinator; in MTT, it is the first conjunct. In addition, MTT allows bracketing of coordinate structure to distinguish different scoping for modifiers. The analyses of WG, FGD, and MTT are all illustrated in **Figure 2**, above, along with the bubble-tree representation proposed by Kahane (1997). More recently, Gerdes & Kahane (2015) have argued that we need to go beyond trees to obtain an adequate analysis of coordination, especially nonconstituent coordination. Their proposal combines symmetrical and asymmetrical analyses into a directed acyclic graph.

Many languages also possess grammaticalized function words like auxiliaries and case markers, which often correspond to morphological inflection (or nothing at all) in other languages. Again, Tesnière (1959) does not assume that such words enter into dependency relations but instead analyzes them using the notions of transfer and dissociated nuclei. Most other dependency grammar theories apply the notion of dependency to function words as well, but there is no universal agreement about the direction of the dependencies, and multistratal theories often treat function words differently in surface syntax and deep syntax. What is characteristic of these structures is that the properties typically associated with syntactic heads are distributed over more than one word. For example, in the case of verb groups, subject–verb agreement is typically marked on the auxiliary (if at all), while valency is determined by the main verb. The status of function words in dependency grammar is discussed by Groß & Osborne (2015), Kahane & Mazziotta (2015), and Osborne (2015), among others. For a historical overview of the treatment of function words in different dependency grammar frameworks, see Osborne & Maxwell (2015).

In dependency parsing, the choice of head can also be influenced by considerations of learnability and parsability, and it has been claimed that parsing is facilitated by treating function words as heads. This appears to be an oversimplification, and recent studies suggest that there is a more complex interaction between representational choices and parsing performance (Schwartz et al. 2012, Rosa 2015, Silveira & Manning 2015, de Lhoneux & Nivre 2016, Rehbein et al. 2017, Wisniewski & Lacroix 2017).

**Figure 4*a*, *b***, and ***c*** illustrates the difference between consistently treating function words as heads, consistently treating content words as heads, and using a Tesnière-style analysis where dependency relations hold between (possibly dissociated) nuclei, respectively. A possible formalization of the third analysis is to use Kahane's (1997) bubble trees, originally proposed for the analysis of coordination.

## 3. THE IMPACT OF DEPENDENCY GRAMMAR IN LINGUISTICS AND NATURAL LANGUAGE PROCESSING

The European syntactic tradition has always been more or less influenced by dependency grammar, and the same is true in other parts of the world, such as India (Bharati & Sangal 1993)

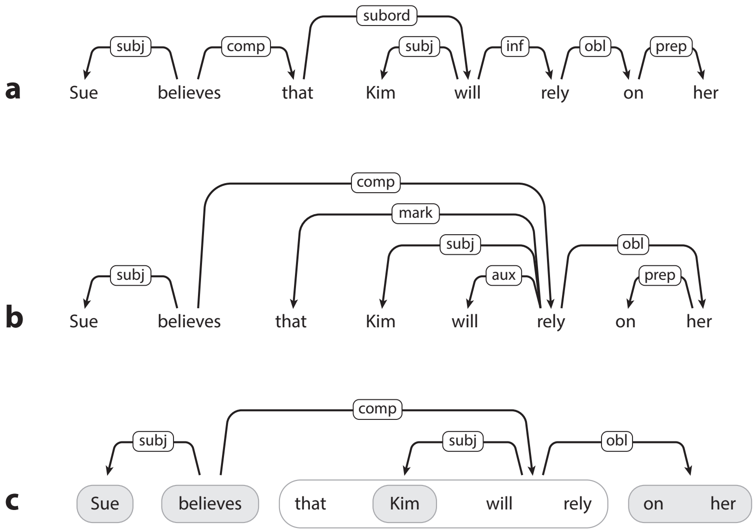


**Figure 4**

Dependency trees with different heads. (*a*) Function words. (*b*) Content words. (*c*) Nuclei à la Tesnière.

and Japan (Uchimoto et al. 1999). While the American syntactic tradition was first rooted in constituency-based frameworks and phrase-structure trees, in both structuralist and generative linguistics, dependency representations have been more widely adopted in recent years, especially in NLP. We highlight here some of the primary reasons for the gain in popularity of dependency representations. These reasons can be viewed as falling under three main advantages that dependency representations offer: generalization across languages, a convenient operationalization of human sentence processing facts, and the transparency and simplicity of the representation.

## 3.1. Generalization Across Languages

Dependency trees are not sensitive to the order of the words in a sentence, in contrast to phrase-structure trees. For instance, both sentences in examples 1*a* and 1*b*, which differ only in the place of the adverbial subordinate clause, at the beginning of the sentence in example 1*a* and at the end of the sentence in example 1*b*, have identical dependency representations, but the phrase-structure trees for these sentences are rather different:

(1a) While it was snowing, I went for a run.

(1b) I went for a run, while it was snowing.

Dependency trees can thus capture generalizations better in languages with free or flexible word order than phrase-structure trees (and keeping track of the surface realization is not particularly difficult, since the word order can be kept by indexing the words following the surface realization). Dependency representations have therefore been favored in annotation efforts for many languages. Some early examples are dependency treebanks for Dutch (Bouma et al. 2000), Czech (Hajič et al. 2001), Danish (Kromann 2003), and Turkish (Oflazer et al. 2003).

Dependency representations are also appealing from the point of view of linguistic typology. There is a long tradition in typology that uses grammatical functions as comparative concepts for crosslinguistic comparison (Greenberg 1963, Keenan & Comrie 1977, Stassen 1985, Croft 2002). Croft et al. (2017) have recently shown how dependency representations are well suited to capture

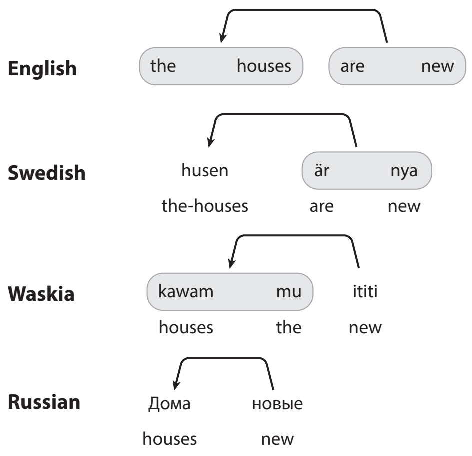


**Figure 5**

Strategies for expressing nonverbal predication (and definiteness).

differences in morphosyntactic realization of universal constructions across languages. We use the example of nonverbal predication to illustrate the different strategies that languages can use to express the same construction. **Figure 5** shows sentences in four languages where the property of being new is predicated of a set of houses, and where all four languages use slightly different morphosyntactic realization strategies. Some languages (e.g., English and Swedish) use a copula strategy (represented as a single nucleus in **Figure 5**), whereas other languages (e.g., Waskia, a language of Papua New Guinea, and Russian) employ a zero strategy in which only the argument and the predicate are overtly expressed. Furthermore, languages differ in how a property like definiteness is conveyed: English and Waskia use a function word to express it (again represented as a nucleus together with the noun in **Figure 5**), while definiteness is morphologically marked in Swedish, and Russian does not mark definiteness at all. Thus, the four languages in **Figure 5** display strategies that vary along two axes, copula versus zero strategy; definiteness is marked with a determiner, marked with morphological inflection, or not marked at all. **Figure 5** displays the four possible combinations, and the dependency representation with bubbles (two bubbles versus one or zero bubbles) makes the differences salient. These differences in strategy, as well as the similarity in constructions, are thus clearly demonstrated by a dependency representation, but would be harder to see in a phrase-structure representation.

Similarly, the notion of head and dependent, which is central to dependency grammar, has been used to categorize languages from a typological perspective. For instance, Nichols (1986) investigated morphological marking and cataloged languages with respect to where the marking appears. Syntactic relations can be morphologically marked either on the head or on the dependent, or on both or neither. This simple descriptive fact, which has implications for typology, historical linguistics, and grammatical theory, emerges clearly when a dependency representation is used. Other studies that make use of dependencies and head-dependent patterns to classify languages include those by Liu (2010), Futrell et al. (2015b), and Chen & Gerdes (2017).

## 3.2. Operationalization of Human Sentence Processing Facts

Another domain where dependency representations have proven very useful is the field of psycholinguistics. For sentence production, dependency representations have been used to study factors that govern the ordering of phrases in a sentence. For sentence comprehension, they have been used to explain why certain structures are more difficult to process.

It has long been suggested (as early as Behaghel 1932) that languages tend to place related words close together. Given a dependency representation, this idea can be formalized as languages exhibiting a preference for shorter dependencies (Temperley 2007). For instance, why do people prefer example 2*a* to 2*b* but exhibit no strong preference for example 3*a* over 3*b*?

(2a) She threw out the bin with old trash.

(2b) She threw the bin with old trash out.

(3a) She threw out the bin.

(3b) She threw the bin out.

The preference for example 2*a* over 2*b* can be explained in terms of dependency length (the number of words between a dependent and its head). Both *out* and *bin* are dependents of *threw*; in example 2*a*, they are both relatively close to their head, but in example 2*b*, there is a much longer dependency between *threw* and *out* because of the length of the noun phrase headed by *bin*. (In examples 3*a* and 3*b*, there is no such difference because the noun phrase is shorter.) This language preference is referred to as dependency length minimization (see Temperley & Gildea 2018 and references therein). Studies in this vein include those by Liu (2008), Futrell et al. (2015a), Gulordava et al. (2015), and Kahane et al. (2017).

Dependencies, and the idea of dependency length minimization, have also been used to explain processing difficulties. For instance, Gibson (1998, 2000) uses the notion of dependency to explain why object relative clauses, as in example 4*a* (Gibson 1998, p. 2), are more difficult to process than subject relative clauses, as in example 4*b*. In example 4*a*, *who*, a dependent of the verb *attacked*, is further away from its head than in example 4*b*:

(4a) The reporter who the senator attacked admitted the error.

(4b) The reporter who attacked the senator admitted the error.

Dependencies have thus been a useful representation to operationalize measures that can be linked to language ordering preferences and to syntactic configurations that pose processing difficulties.

## 3.3. Transparency and Simplicity of the Representation

Dependency trees provide a transparent encoding of predicate–argument structure, which in turn easily supports semantic interpretation. They are therefore popular for downstream applications in NLP that focus on predicate–argument relations, such as relation extraction and question answering. Any subtree of a dependency tree is easily interpretable. For instance, from **Figure 1** one can easily extract the information that the subject of *chase* is *dogs* and that *happily* modifies the predicate *chase*. Furthermore, dependency representations are more robust than phrase-structure trees: Even if a parser fails to provide a complete parse for a given sentence, fragments of the dependency tree are still interpretable thanks to the functional labels on dependency arcs.

As pointed out by Covington (2001), the constrained nature of dependency trees makes the parsing task simpler. Constrained representations make it conceptually easier to implement efficient parsers. The input sentence fixes the number of nodes that the tree needs to contain (given that dependency trees contain one node per word), and only these nodes need to be connected to one another. Thus, in dependency parsing, the job of the parser is to link existing nodes together, and not to postulate new ones; by contrast, in constituency parsing, phrase-level nodes need to be inferred. The constrained nature of the dependency parsing task sometimes also permits the use of efficient algorithms that are not applicable to other representations. A prime example is the use of spanning tree algorithms from graph theory, pioneered by McDonald et al. (2005b), to enable nonprojective dependency parsing in quadratic time; the corresponding parsing problem for constituency representations with discontinuities is not even computable in polynomial time. Parsing with dependency grammars was explored as early as the 1960s by Hays (1964), Gaifman (1965), and Robinson (1970), among others, but only for restricted classes of context-free grammars.

Finally, the core structure of dependency trees, that is, binary relations between lexical elements forming a tree, is a conceptually simple representation, and the idea of grammatical relations holding between words is a notion that many people not especially versed in linguistics are nonetheless familiar with. Starosta (1990, p. 86) wrote the following about the appeal of case grammar:

> At the entry level at least anybody can do it [= case grammar], in the privacy of their own home, with no linguistic training required! [...] The reason that anyone can do this kind of case grammar is that football fans for example are very knowledgeable about Goals, aficionados of "General Hospital" or "St. Elsewhere" know all about Patients, and anyone who has watched a James Bond movie can identify an Agent.

Such an explanation can easily be extended to dependency representations using grammatical relations. Anyone can grasp the notion of subject and object and understand that some words can be modified by others. Dependency grammar thus offers a representation that is usable by anyone who wants to build or use systems for text understanding, not only (computational) linguists but also computer scientists more generally, and by information professionals including biologists, medical researchers, political scientists, business and market analysts, and legal professionals.

### 3.4. Limitations of Dependency Representations

While constrained representations have advantages, as described above, they also come with limitations. By necessity, more constrained representations are less expressive, and dependency representations are therefore often underspecified in comparison with more expressive representations. In particular, it is not possible to mark the distinction between an element modifying the head of the phrase and the same element modifying the whole phrase. In **Figure 1**, we cannot distinguish between the adverb modifying only the predicate *chase* and that modifying the predicate and its object *chase cats*. This limitation is especially problematic for dependency frameworks that analyze coordination by treating one of the conjuncts as the syntactic head, such as the MTT analysis in **Figure 2**. It is therefore impossible to structurally distinguish different scopes of modifiers such as *young* in *young men and women*, which is why MTT resorts to the use of bracketing in such cases. Another problematic case, discussed by Mel'čuk (2003), is layered modification, such as *expensive Japanese cars* (in contrast to *Japanese expensive cars*), where the outermost adjective *expensive* appears to modify the phrase *Japanese cars* rather than just the head noun *cars*. In a plain dependency tree, however, both adjectives have to be attached as direct dependents of the noun.

Even if we try to introduce a distinction between words and phrases in dependency trees (for example, by using distinct labels for word and phrase modification), the resulting notion of phrase is very limited since it is reduced to complete subtrees dominated by a word, making it impossible to distinguish different projection levels, as is possible in a phrase-structure tree. For this reason, linguists have explored alternative substructures in dependency trees, which can be used to capture generalizations that cannot be stated (only) in terms of words and subtrees. In particular, the notion of catena, or chain, which is a connected substructure of the dependency tree, has been employed for the analysis of gapping and idioms, among other things (Osborne 2005, Osborne et al. 2012).

Having trees where the nodes are fixed by the input itself also means that empty nodes are not favored. Some linguistic phenomena, such as ellipsis in example 5, are therefore more difficult to analyze adequately, without adding complexity to the representation (for an overview of approaches to ellipsis in dependency grammar, see Hajičová et al. 2015):

(5) Peter is her older brother, Oliver her youngest.

Overall, however, the increasing popularity of dependency representations suggests that they strike a good balance in the trade-off between the expressivity and complexity of linguistic representations.

## 4. UNIVERSAL DEPENDENCIES: A FRAMEWORK FOR MORPHOSYNTACTIC ANNOTATION

The increasing use of dependency representations in linguistics and NLP is clearly visible in corpus annotation. The last decade has seen the emergence of dependency corpora for many languages, potentially enabling crosslinguistic empirical research and computational applications. A major problem, however, is that annotation schemes vary enormously across languages. This variation is illustrated in **Figure 6**, which shows three parallel sentences in Swedish, Danish, and English, annotated according to the guidelines of the Swedish Treebank (Nivre & Megyesi 2007), the Danish Dependency Treebank (Kromann 2003), and Stanford Typed Dependencies (de Marneffe et al. 2006), respectively. While the syntactic structures are identical in the three languages, the percentage of shared dependency relations across pairs of languages is at most 40% (and 0% across all three languages), making it very hard to compare the dependency trees in a meaningful way.

UD (see **http://universaldependencies.org/**) is a recent initiative that tries to deal with this problem by developing a framework for crosslinguistically consistent morphosyntactic annotation (Nivre et al. 2016). The UD framework, which so far has been applied to more than 60 languages, aims to capture similarities and idiosyncrasies among typologically different languages, for example, morphologically rich languages, prodrop languages, and languages featuring clitic doubling. The goal is to support multilingual research in NLP and linguistics by enabling sound comparative

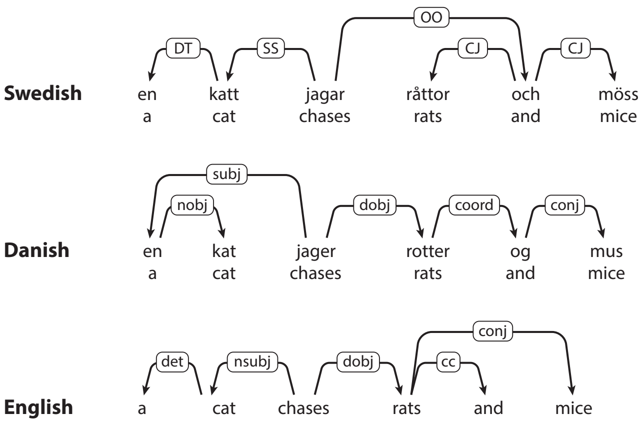


**Figure 6**

Dependency trees for parallel sentences in Swedish, Danish, and English.

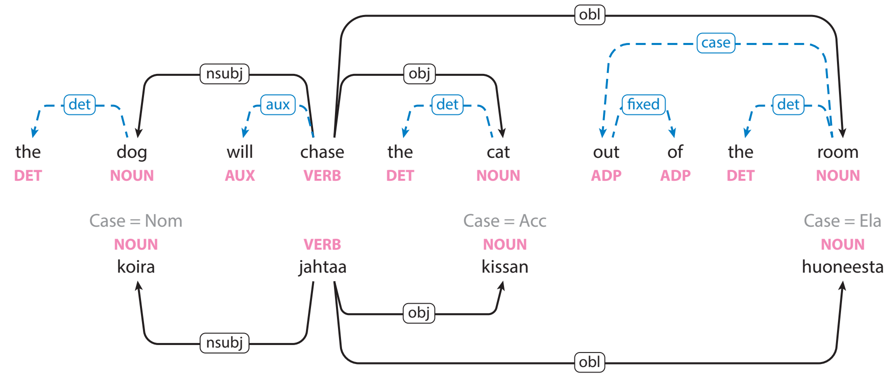


**Figure 7**

Simplified Universal Dependencies annotation for equivalent sentences from English (*top*) and Finnish (*bottom*). Solid black arcs represent dependency relations between content words; dashed blue arcs represent relations between content and function words.

evaluation across languages, crosslinguistic learning to support low-resource languages, development of multilingual natural language understanding systems, and comparative linguistic studies. The UD annotation scheme includes both morphology and syntax, but we focus here on syntactic annotation.

**Figure 7** shows the UD analysis for a sentence in English and its equivalent in Finnish, two languages that are rather different typologically. Both sentences have the same syntactic structure (a predicate with a subject, an object, and a locative modifier) but differ in their morphosyntactic realizations. While English uses function words (determiners, auxiliaries, and prepositions) and word order to encode grammatical relations and categories, Finnish uses case markers (and does not explicitly encode definiteness or future tense). The core dependency structure, however, is parallel between the two sentences. This parallelism is captured in UD by giving priority to dependency relations between content words, while function words are attached to the content word with which they form a nucleus in Tesnière's sense (**Figure 7**).

UD builds on existing de facto standards in the NLP community and therefore constitutes an eclectic framework. The dependency annotation derives from the (universal) Stanford Dependencies (de Marneffe & Manning 2008, de Marneffe et al. 2014), and the morphological annotation combines a revised version of the Google universal part-of-speech tags (Petrov & McDonald 2012) with a set of morphological features from the Interset interlingua for morphosyntactic tag sets (Zeman 2008). UD has also been influenced by several attempts to achieve consistent annotation across languages: HamleDT (Zeman et al. 2012, 2014; Rosa et al. 2014), the Universal Dependency Treebank Project (McDonald et al. 2013), and the unified morphosyntactic Stanford scheme proposed by Tsarfaty (2013). In the following subsections, we characterize the UD framework in relation to the five issues introduced in Section 2.

## 4.1. Sufficiency of Dependency

The question of whether UD assumes dependency to be sufficient as well as necessary for a complete syntactic analysis can be answered in two different ways. Formally speaking, UD does reduce syntactic structure to directed binary relations between lexical elements, relations that form a spanning tree over the words of the sentence, and does not allow notions of constituency or bracketing in the trees. So, in this sense, UD uses "pure" dependency representations.

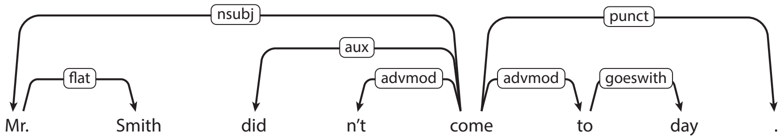


**Figure 8**

Universal Dependencies dependency tree displaying nonsyntactic relations *flat* and *goeswith*.

However, UD does not assume that all binary relations in the tree represent relations between a syntactic head and a dependent. Thus, whereas argument and modifier relations like *nsubj*, *obj*, and *obl* in **Figure 7** are regarded as syntactic dependency relations, functional relations like *det*, *aux*, and *case* are considered to be relations holding between a lexical head and its grammatical markers. In this sense, UD is closer to the older tradition of Tesnière (1959) in assuming that some lexical elements form a nucleus with no clear syntactic head, and therefore does not reduce everything to dependency in the narrow theoretical sense.

In addition, UD trees contain relations that are not syntactic relations at all, such as *flat* for exocentric constructions like title–name combinations (e.g., *Mr. Smith* in **Figure 8**) and *goeswith* for accidentally split words in nonedited text (as *to day* for *today* in **Figure 8**). Clearly, not all arcs from a lexical element $x$ to a lexical element $y$ in a UD tree can be interpreted as meaning that $y$ is a syntactic dependent of $x$.

## 4.2. Levels of Representation

In the case of UD, the question about levels of representation does not have a definitive answer. To date, most research has been concerned with a single layer of representation known as "basic" dependencies, where the dependency structure is a spanning tree over the words of the sentence that encodes surface syntactic relations. This is the only representation that is available in the majority of UD treebanks, which are therefore best described as monostratal.

However, the UD guidelines also define a second layer of representation, known as "enhanced" dependencies, which encodes deeper relations that are potentially useful for natural language understanding systems, such as implicit subjects in raising and control constructions, as well as some coreference relations, in particular in relative clauses. Although many dependency relations are common to the basic and enhanced representations, they are in principle independent structures, which in some sense makes UD a multistratal framework.

Both representations are illustrated in **Figure 9**, with the basic representation above the sentence and the enhanced one below it. For the relative clause, the enhanced dependencies add a coreference link (*ref*) from the antecedent *students* to the relative pronoun *who* and a deep subject relation (*nsubj*) from *forgot* to *students*. The subject in the control construction is indicated by the relation *nsubj* from the verb *come* to *students*. Finally, the adjectival modifier relation in the coordination *oral and written* is propagated. Both adjectives in the coordination modify the noun *exam*, which is indicated by the additional *amod* relation from *exam* to *written*.

## 4.3. Lexical Elements and Relations

UD adopts a lexicalist view of syntax and takes syntactic words as the basic units of annotation (and not phonological or orthographic words). Thus, clitics are separated and contractions are undone. For instance, *dámelo* 'give-me-it' in Spanish is treated as three words: *da me lo*. Morphological features, by contrast, are encoded as properties of words and there is no attempt at segmenting

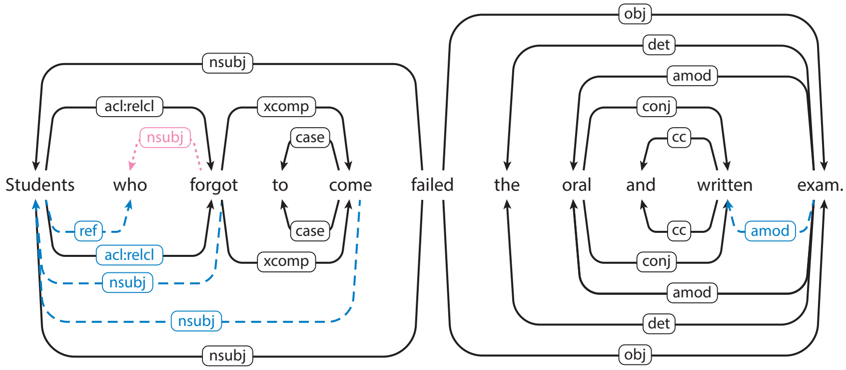


**Figure 9**

Universal Dependencies basic dependency tree (*top*) and enhanced dependency graph (*bottom*). Solid black arcs are shared across both representations. Dotted pink arcs appear only in the basic representation, whereas dashed blue arcs appear only in the enhanced representation.

words into morphemes (as shown in **Figure 7**, where the case markers in Finnish are not split). In the basic dependency representation, syntactic words have a one-to-one correspondence to nodes of the dependency tree. In the enhanced representation, there can also be empty nodes that do not correspond to actual words. So far, empty nodes are used only for elided predicates in gapping constructions.

Regarding relations, UD uses a taxonomy of 37 universal relations, which may be further specified by subtype. The core of the taxonomy is based on traditional grammatical functions like subject and object and is organized by two main principles. The first is a division into three main types of linguistic structures: clauses, nominals, and modifier words. This division is reflected in a systematic distinction between nominal and clausal subjects (*nsubj* versus *csubj*), nominal and clausal objects (*obj* versus *xcomp* or *ccomp*), phrasal and clausal adverbials (*advmod* versus *advcl*), and so on. It is also reflected in the use of different relations for nominal modifiers at the level of clauses (*obl*) and the level of nominals (*nmod*). The second principle is a distinction between core arguments, essentially subjects and objects, and oblique modifiers at the clause level.

In addition to basic grammatical functions, the UD taxonomy includes special relations for function word relations (e.g., *case* and *aux*, discussed above) and for phenomena such as coordination, list structures, ellipsis, disfluencies, and restricted classes of multiword expressions. As noted above, some of these relations are not really dependency relations in the narrow sense, but rather are devices for encoding special phenomena in the shape of a tree. A typical example is the *fixed* relation, which is used to connect the subcomponents of a fixed expression like *as well as* without making any claims about internal syntactic structure.

The choice of relations for UD is motivated by the desire to satisfy a number of (partly conflicting) requirements for the use of UD for annotation. The framework should enable satisfactory linguistic analysis for individual languages but should also reveal crosslinguistic parallelism across languages and language families. It should be suitable for rapid, consistent annotation by a human annotator, as well as for computer parsing with high accuracy. The relations in UD should be easily comprehended and used by a nonlinguist, whether a language learner or an engineer with prosaic needs for language processing, which leads to a preference for traditional grammar notions and terminology. Finally, the relations offered by UD should be useful for downstream language understanding tasks in NLP.

## 4.4. Formal Properties of Dependency Representations

When discussing formal properties of UD representation, we again need to distinguish basic and enhanced dependencies. The basic dependency representation is a spanning tree over the words of the sentence, as defined in Section 2. The enhanced dependency representation is a general graph structure, where words may have more than one incoming arc and where there may even be cycles in the graph. In addition, as mentioned above, the enhanced representation may contain empty nodes that do not correspond to a word of the sentence.

## 4.5. Criteria for Dependency Relations

When it comes to criteria for distinguishing dependency relations—and syntactic heads in those relations—UD follows the mainstream tradition for the core of uncontroversial constructions discussed in Section 2. Thus, arguments of predicates and modifiers in endocentric constructions are consistently treated as syntactic dependents at both the clausal and phrasal levels.

For other types of constructions, UD may appear to go against the mainstream tradition, at least for surface syntactic representations. This is true in particular for relations between content words and function words, where UD consistently puts content words higher in the trees, while most dependency grammar frameworks treat the function words as syntactic heads (Osborne & Maxwell 2015). However, as discussed above, the content word together with the attached function word(s) can be understood as the representation of a dissociated nucleus in Tesnière's sense, a structure that cannot be reduced to dependency relations.

Similarly, UD represents coordination by linking all subsequent conjuncts to the first conjunct using the *conj* relation, but this should again be understood as a tree encoding of a flat structure without a syntactic head. The same rigid left-to-right linking is found in constructions that have no internal syntactic structure at all, such as fixed multiword expressions (*fixed*), as well as loosely joined paratactic constructions and lists (*parataxis*, *list*). The apparent conflict between UD and mainstream dependency grammar can thus be explained by the fact that UD uses the formal dependency tree representation to encode a multitude of relations, many of which have no clear syntactic heads. The decision in UD of putting content words higher in the trees than function words makes the framework more appealing as a representation for linguistic typology (Croft et al. 2017), since it produces parallel representations of constructions that are realized by different strategies in different languages, as discussed in Section 3.

# 5. CONCLUSION

In this article, we have reviewed dependency grammar from several perspectives, trying to connect its original development in linguistics to its more recent use in NLP, with the development of dependency-annotated corpora as a bridge between the two. It seems fair to say that the use of dependency structures in NLP has to a great extent remained uninformed by the theoretical tradition of dependency grammar and has consisted mainly of borrowing a formal representation that strikes a good balance between expressivity and complexity, as discussed in Section 3. In addition, most research on dependency parsing has been limited to syntax, whereas the notion of dependency in linguistics has also been applied to morphology and semantics. This situation may be changing, and we have definitely witnessed a growing interest in modeling semantics using dependencies in recent years (Oepen et al. 2014, 2015). We have also seen increasing interaction between theoretical and applied approaches to dependency grammar, in particular through a new conference series on dependency linguistics (Gerdes et al. 2011, Hajičová et al. 2013, Hajičová &

Nivre 2015, Montemagni & Nivre 2017), which specifically aims to bring different dependency grammar communities together.

## DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.



## LITERATURE CITED

Attardi G. 2006. Experiments with a multilanguage non-projective dependency parser. In *Proceedings of the 10th Conference on Computational Natural Language Learning*, pp. 166–70. Stroudsburg, PA: Assoc. Comput. Linguist.

Behaghel O. 1932. *Deutsche Syntax: Eine geschichtliche Darstellung*. Band IV: *Wortstellung-periodenbau*. Heidelberg, Ger.: Carl Winter

Bharati A, Sangal R. 1993. Parsing free word order languages in the Paninian framework. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 105–11. Stroudsburg, PA: Assoc. Comput. Linguist.

Bouma G, van Noord G, Malouf R. 2000. Alpino: wide-coverage computational analysis of Dutch. In *Computational Linguistics in the Netherlands 2000: Selected Papers from the 11th CLIN Meeting*, pp. 45–59. Leiden, Neth.: Brill

Chen X, Gerdes K. 2017. Classifying languages by dependency structure: typologies of delexicalized Universal Dependency treebanks. In *Proceedings of the 4th International Conference on Dependency Linguistics*, pp. 54–63. Stroudsburg, PA: Assoc. Comput. Linguist.

Chomsky N. 1970. Remarks on nominalization. In *Readings in English Transformational Grammar*, ed. RA Jacobs, PS Rosenbaum, pp. 11–61. Boston: Ginn

Chomsky N. 1998. *Minimalist Inquiries: The Framework*. Cambridge, MA: MIT Press

Covington MA. 1984. *Syntactic Theory in the High Middle Ages*. Cambridge, UK: Cambridge Univ. Press

Covington MA. 2001. A fundamental algorithm for dependency parsing. In *Proceedings of the 39th Annual ACM Southeast Conference*, pp. 95–102. Athens: Univ. Ga.

Croft W. 2002. *Typology and Universals*. Cambridge, UK: Cambridge Univ. Press. 2nd ed.

Croft W, Nordquist D, Looney K, Regan M. 2017. Linguistic typology meets Universal Dependencies. In *Proceedings of the 15th International Workshop on Treebanks and Linguistic Theories*, pp. 63–75. Bloomington: Univ. Indiana

de Lhoneux M, Nivre J. 2016. Should have, would have, could have: investigating verb group representations for parsing with Universal Dependencies. In *Proceedings of the Workshop on Multilingual and Cross-Lingual Methods in NLP*, pp. 10–19. Stroudsburg, PA: Assoc. Comput. Linguist.

de Marneffe MC, Dozat T, Silveira N, Haverinen K, Ginter F, et al. 2014. Universal Stanford Dependencies: a cross-linguistic typology. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pp. 26–31. Paris: Eur. Lang. Resour. Assoc.

de Marneffe MC, MacCartney B, Manning CD. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pp. 449–54. Paris: Eur. Lang. Resour. Assoc.

de Marneffe MC, Manning CD. 2008. The Stanford typed dependencies representation. In *Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation* (*COLING 2008*), pp. 1–8. Stroudsburg, PA: Assoc. Comput. Linguist.

Dowty D. 1989. On the semantic content of the notion of 'thematic role.' In *Properties, Types and Meaning*, vol. 2: *Semantic Issues*, ed. G Chierchia, BH Partee R Turner, pp. 69–130. Dordrecht, Neth.: Reidel

Eisner JM. 1996. Three new probabilistic models for dependency parsing: an exploration. In *Proceedings of the 16th International Conference on Computational Linguistics* (*COLING96*), pp. 340–45. Stroudsburg, PA: Assoc. Comput. Linguist.

Fillmore CJ. 1968. The case for case. In *Universals in Linguistic Theory*, ed. EW Bach, RT Harms, pp. 1–88. New York: Holt, Rinehart & Winston

Futrell R, Mahowald K, Gibson E. 2015a. Large-scale evidence of dependency length minimization in 37 languages. *PNAS* 112:10336–41

Futrell R, Mahowald K, Gibson E. 2015b. Quantifying word order freedom in dependency corpora. See Hajičová & Nivre 2015, pp. 91–100

Gaifman H. 1965. Dependency systems and phrase-structure systems. *Inform. Control* 8:304–37

Gerdes K, Hajičová E, Wanner L, ed. 2011. *Proceedings of the 1st International Conference on Dependency Linguistics*. Stroudsburg, PA: Assoc. Comput. Linguist.

Gerdes K, Kahane S. 2015. Nonconstituent coordination and other coordinative constructions as dependency graphs. See Hajičová & Nivre 2015, pp. 101–10

Gibson E. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition* 68:1–76

Gibson E. 2000. The dependency locality theory: a distance-based theory of linguistic complexity. In *Images, Language, Brain: Papers from the 1st Mind Articulation Symposium*, ed. AP Marantz, Y Miyashita, W O'Neil, pp. 95–126. Cambridge, MA: MIT Press

Gómez-Rodríguez C, Nivre J. 2010. A transition-based parser for 2-planar dependency structures. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1492–501. Stroudsburg, PA: Assoc. Comput. Linguist.

Greenberg JH. 1963. Some universals of grammar with particular reference to the order of meaningful elements. In *Universals of Human Language*, ed. JH Greenberg, pp. 73–113. Cambridge, MA: MIT Press

Groß T, Osborne T. 2015. The dependency status of function words: auxiliaries. See Hajičová & Nivre 2015, pp. 111–20

Gulordava K, Merlo P, Crabbé B. 2015. Dependency length minimisation effects in short spans: a large-scale analysis of adjective placement in complex noun phrases. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, vol. 2: *Short Papers*, pp. 477–82. Stroudsburg, PA: Assoc. Comput. Linguist.

Hajič J, Hladka BV, Panevová J, Hajičová E, Sgall P, Pajas P. 2001. *Prague Dependency Treebank, version 1.0*. Czech–English corpus. CD-ROM LDC2001T10

Hajičová E, Gerdes K, Wanner L, ed. 2013. *Proceedings of the 2nd International Conference on Dependency Linguistics*. Stroudsburg, PA: Assoc. Comput. Linguist.

Hajičová E, Mikulová M, Panevová J. 2015. Reconstructions of deletions in a dependency-based description of Czech: selected issues. See Hajičová & Nivre 2015, pp. 131–40

Hajičová E, Nivre J, ed. 2015. *Proceedings of the 3rd International Conference on Dependency Linguistics*. Stroudsburg, PA: Assoc. Comput. Linguist.

Hays DG. 1964. Dependency theory: a formalism and some observations. *Language* 40:511–25

Hudson RA. 1984. *Word Grammar*. Oxford, UK: Blackwell

Hudson RA. 1990. *English Word Grammar*. Oxford, UK: Blackwell

Hudson RA. 2007. *Language Networks: The New Word Grammar*. Oxford, UK: Oxford Univ. Press

Jackendoff R. 1972. *Semantic Interpretation in Generative Grammar*. Cambridge, MA: MIT Press

Jackendoff R. 1977. $\overline{X}$ *Syntax: A Study of Phrase Structure*. Cambridge, MA: MIT Press

Joshi A. 1985. How much context-sensitivity is necessary for characterizing structural descriptions—tree adjoining grammars. In *Natural Language Processing: Psycholinguistic, Computational and Theoretical Perspectives*, ed. D Dowty, L Karttunen, A Zwicky, pp. 206–50. Cambridge, UK: Cambridge Univ. Press

Kahane S. 1997. Bubble trees and syntactic representations. In *Proceedings of the 5th Meeting of Mathematics of Language*, pp. 70–76. Saarbrücken, Ger.: Dtsch. Forsch. Künstliche Intell.

Kahane S, Mazziotta N. 2015. Dependency-based analyses for function words—introducing the polygraphic approach. See Hajičová & Nivre 2015, pp. 181–90

Kahane S, Yan C, Botalla MA. 2017. What are the limitations on the flux of syntactic dependencies? Evidence from UD treebanks. See Montemagni & Nivre 2017, pp. 73–82

Keenan EL, Comrie B. 1977. Noun phrase accessibility and universal grammar. *Linguist. Inq.* 8:63–99

Koo T, Collins M. 2010. Efficient third-order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1–11. Stroudsburg, PA: Assoc. Comput. Linguist.

Koo T, Rush AM, Collins M, Jaakkola T, Sontag D. 2010. Dual decomposition for parsing with non-projective head automata. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (*EMNLP10*), pp. 1288–98. Stroudsburg, PA: Assoc. Comput. Linguist.

Kromann MT. 2003. The Danish Dependency Treebank and the DTAG treebank tool. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories*, pp. 217–22. Väsjö, Swed.: Väsjö Univ. Press

Kruijff GJM. 2002. *Formal and computational aspects of dependency grammar: history and development of DG*. Tech. rep. ESSLLI-2002, Univ. Saarland, Saarbrücken, Ger.

Kübler S, McDonald R, Nivre J. 2009. *Dependency Parsing*. San Rafael, CA: Morgan & Claypool

Kuhlmann M, Möhl M. 2007. Mildly context-sensitive dependency languages. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 160–67. Stroudsburg, PA: Assoc. Comput. Linguist.

Lecerf Y. 1960. Programme des conflits, modèle des conflits. *Bull. Bimest. l'ATALA* 1(4):11–18, 1(5):18–36

Liu H. 2008. Dependency distance as a metric of language comprehension difficulty. *J. Cogn. Sci.* 9:159–91

Liu H. 2010. Dependency direction as a means of word-order typology: a method based on dependency treebanks. *Lingua* 120:1567–78

Marcus S. 1965. Sur la notion de projectivité. *Z. Math. Log. Grundl. Math.* 11:181–92

Martins A, Smith N, Xing E. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 342–50. Stroudsburg, PA: Assoc. Comput. Linguist.

Matthews PH. 1981. *Syntax*. Cambridge, UK: Cambridge Univ. Press

McDonald R, Crammer K, Pereira F. 2005a. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 91–98. Stroudsburg, PA: Assoc. Comput. Linguist.

McDonald R, Nivre J, Quirmbach-Brundage Y, Goldberg Y, Das D, et al. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, vol. 2: *Short Papers*, pp. 92–97. Stroudsburg, PA: Assoc. Comput. Linguist.

McDonald R, Pereira F, Ribarov K, Hajič J. 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* (*HLT05*), pp. 523–30. Stroudsburg, PA: Assoc. Comput. Linguist.

Mel'čuk I. 1988. *Dependency Syntax: Theory and Practice*. Albany, NY: SUNY Press

Mel'čuk I. 2003. Levels of dependency in linguistic description: concepts and problems. In *Dependency and Valency: An International Handbook of Contemporary Research*, ed. V Agel, L Eichinnger, HW Eroms, P Hellwig, H Herringer, H Lobin, pp. 188–229. Berlin: Walter de Gruyter

Milicevic J. 2006. A short guide to the Meaning–Text linguistic theory. *J. Koralex* 8:187–233

Montemagni S, Nivre J, ed. 2017. *Proceedings of the 4th International Conference on Dependency Linguistics*. Stroudsburg, PA: Assoc. Comput. Linguist.

Nichols J. 1986. Head-marking and dependent-marking grammar. *Language* 62:56–119

Nivre J. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies*, pp. 149–60. Stroudsburg, PA: Assoc. Comput. Linguist.

Nivre J. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 351–59. Stroudsburg, PA: Assoc. Comput. Linguist.

Nivre J, de Marneffe MC, Ginter F, Goldberg Y, Hajič J, et al. 2016. Universal Dependencies v1: a multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pp. 1659–66. Paris: Eur. Lang. Resour. Assoc.

Nivre J, Hall J, Kübler S, McDonald R, Nilsson J, et al. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pp. 915–32. Stroudsburg, PA: Assoc. Comput. Linguist.

Nivre J, Megyesi B. 2007. Bootstrapping a Swedish treebank using cross-corpus harmonization and annotation projection. In *Proceedings of the 6th Workshop on Treebanks and Linguistic Theories*, pp. 97–102. Bergen, Nor.: North. Eur. Assoc. Lang. Technol.

Nivre J, Nilsson J. 2005. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 99–106. Stroudsburg, PA: Assoc. Comput. Linguist.

Oepen S, Kuhlmann M, Miyao Y, Zeman D, Cinkova S, et al. 2015. SemEval 2015 task 18: broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation* (*SemEval 2015*), pp. 915–26. Stroudsburg, PA: Assoc. Comput. Linguist.

Oepen S, Kuhlmann M, Miyao Y, Zeman D, Flickinger D, et al. 2014. SemEval 2014 task 8: broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation* (*SemEval 2014*), pp. 63–72. Stroudsburg, PA: Assoc. Comput. Linguist.

Oflazer K, Say B, Hakkani-Tür DZ, Tür G. 2003. Building a Turkish treebank. In *Treebanks: Building and Using Parsed Corpora*, ed. A Abeillé, pp. 261–77. Dordrecht, Neth.: Kluwer

Osborne T. 2005. Beyond the constituent: a DG analysis of chains. *Folia Linguist.* 39:251–97

Osborne T. 2015. Diagnostics for constituents: dependency, constituency, and the status of function words. See Hajičová & Nivre 2015, pp. 251–60

Osborne T, Maxwell D. 2015. A historical overview of the status of function words in dependency grammar. See Hajičová & Nivre 2015, pp. 241–50

Osborne T, Putnam M, Groß M. 2012. Catenae: introducing a novel unit of syntactic analysis. *Syntax* 15:354–96

Petrov S, McDonald R. 2012. Overview of the 2012 shared task on parsing the web. In *Notes of the 1st Workshop on Syntactic Analysis of Non-Canonical Language*. 8 pp. **https://sites.google.com/site/sancl2012/home/programme**

Pitler E. 2012. Attacking parsing bottlenecks with unlabeled data and relevant factorizations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, vol. 1: *Long Papers*, pp. 768–76. Stroudsburg, PA: Assoc. Comput. Linguist.

Pitler E, Kannan S, Marcus M. 2013. Finding optimal 1-endpoint-crossing trees. *Trans. Assoc. Comput. Linguist.* 1:13–24

Pollard C, Sag IA. 1987. *Information-Based Syntax and Semantics*. Stanford, CA: Cent. Study Lang. Inf.

Pollard C, Sag IA. 1994. *Head-Driven Phrase Structure Grammar*. Stanford, CA: Cent. Study Lang. Inf.

Rehbein I, Steen J, Do J, Frank A. 2017. Universal dependencies are hard to parse—or are they? See Montemagni & Nivre 2017, pp. 218–28

Robinson JJ. 1970. Dependency structures and transformational rules. *Language* 46:259–85

Rosa R. 2015. Multi-source cross-lingual delexicalized parser transfer: Prague or Stanford? See Hajičová & Nivre 2015, pp. 281–90

Rosa R, Mašek J, Mareçek D, Zeman D, Žabkrtský Z. 2014. HamleDT 2.0: thirty dependency treebanks Stanfordized. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pp. 2334–41. Paris: Eur. Lang. Resour. Assoc.

Schwartz R, Abend O, Rappoport A. 2012. Learnability-based syntactic annotation design. In *Proceedings of the 24th International Conference on Computational Linguistics* (*COLING 2012*), pp. 2405–21. Stroudsburg, PA: Assoc. Comput. Linguist.

Sgall P, Hajičová E, Panevová J. 1986. *The Meaning of the Sentence in Its Pragmatic Aspects*. Dordrecht, Neth.: Reidel

Silveira N, Manning C. 2015. Does Universal Dependencies need a parsing representation? An investigation of English. See Hajičová & Nivre 2015, pp. 310–19

Starosta S. 1990. Review of H. Somers: *Valency and Case in Computational Linguistics*. Mach. Transl. 5:79–96

Stassen L. 1985. *Comparison and Universal Grammar*. Oxford, UK: Blackwell

Temperley D. 2007. Minimization of dependency length in written English. *Cognition* 105:300–33

Temperley D, Gildea D. 2018. Minimizing syntactic dependency lengths: typological/cognitive universal? *Annu. Rev. Linguist.* 4:1–15

Tesnière L. 1959. *Éléments de syntaxe structurale*. Paris: Ed. Klincksieck

Tesnière L. 2015. *Elements of Structural Syntax*, transl. T Osborne, S Kahane. Amsterdam: Benjamins

Tsarfaty R. 2013. A unified morpho-syntactic scheme of Stanford dependencies. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, vol. 2: *Short Papers*, pp. 578–84. Stroudsburg, PA: Assoc. Comput. Linguist.

Uchimoto K, Sekine S, Isahara H. 1999. Japanese dependency structure analysis based on maximum entropy models. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics* (*EACL 99*), pp. 196–203. Stroudsburg, PA: Assoc. Comput. Linguist.
Wisniewski G, Lacroix O. 2017. A systematic comparison of syntactic representations of dependency parsing. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies*, pp. 146–52. Stroudsburg, PA: Assoc. Comput. Linguist.
Yamada H, Matsumoto Y. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies*, pp. 195–206. Stroudsburg, PA: Assoc. Comput. Linguist.
Zeman D. 2008. Reusable tagset conversion using tagset drivers. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, pp. 213–18. Paris: Eur. Lang. Resour. Assoc.
Zeman D, Dušek O, Popel M, Ramasamy L, Štěpánek J, et al. 2014. HamleDT: harmonized multi-language dependency treebank. *Lang. Resour. Eval.* 48:601–37
Zeman D, Mareček D, Popel M, Ramasamy L, Štěpánek J, et al. 2012. HamleDT: to parse or not to parse? In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pp. 2735–41. Paris: Eur. Lang. Resour. Assoc.
Zwicky AM. 1985. Heads. *J. Linguist.* 21:1–29

Annual Review of Linguistics

Volume 5, 2019

# Contents

**Errata**

An online log of corrections to *Annual Review of Linguistics* articles may be found at http://www.annualreviews.org/errata/linguistics