Nucleus Composition in Transition-based Dependency Parsing

Joakim Nivre
Uppsala University
Department of Linguistics and Philology
RISE Research Institutes of Sweden
joakim.nivre@lingfil.uu.se

Ali Basirat
Linköping University
Department of Computer and
Information Science
ali.basirat@liu.se

Luise Dürlich
Uppsala University
Department of Linguistics and Philology
RISE Research Institutes of Sweden
luise.durlich@ri.se

Adam Moss
Uppsala University
Department of Linguistics and Philology
adam.moss@lingfil.uu.se

Dependency-based approaches to syntactic analysis assume that syntactic structure can be analyzed in terms of binary asymmetric dependency relations holding between elementary syntactic units. Computational models for dependency parsing almost universally assume that an elementary syntactic unit is a word*, while the influential theory of Lucien Tesnière instead posits a more abstract notion of* nucleus*, which may be realized as one or more words. In this article, we investigate the effect of enriching computational parsing models with a concept of nucleus inspired by Tesnière. We begin by reviewing how the concept of nucleus can be defined in the framework of Universal Dependencies, which has become the de facto standard for training and evaluating supervised dependency parsers, and explaining how composition functions can be used to make neural transition-based dependency parsers aware of the nuclei thus defined. We then perform an extensive experimental study, using data from 20 languages to assess the impact*

Action Editor: Carlos Gómez-Rodríguez. Submission received: 28 February 2022; revised version received: 10 June 2022; accepted for publication: 23 June 2022.

https://doi.org/10.1162/coli_a_00450

of nucleus composition across languages with different typological characteristics, and utilizing a variety of analytical tools including ablation, linear mixed-effects models, diagnostic classifiers, and dimensionality reduction. The analysis reveals that nucleus composition gives small but consistent improvements in parsing accuracy for most languages, and that the improvement mainly concerns the analysis of main predicates, nominal dependents, clausal dependents, and coordination structures. Significant factors explaining the rate of improvement across languages include entropy in coordination structures and frequency of certain function words, in particular determiners. Analysis using dimensionality reduction and diagnostic classifiers suggests that nucleus composition increases the similarity of vectors representing nuclei of the same syntactic type.

1. Introduction

A syntactic analysis in the form of a dependency tree consists of labeled directed arcs, which represent grammatical relations like subject and object. These arcs connect a set of nodes, which represent the basic syntactic units of a sentence. Standard models of dependency parsing generally assume that the elementary units are tokens or word forms, which are the output of a tokenizer or word segmenter. This assumption gives rise to considerable variation in the shape and size of dependency trees across languages, because of different typological characteristics. Thus, morphologically rich languages typically have fewer elementary units and fewer relations than more analytical languages, which to a larger extent rely on function words instead of morphological inflection to encode grammatical information. This variation is illustrated in Figure 1, which compares two equivalent sentences in English and Finnish, annotated with dependency trees following the guidelines of Universal Dependencies (UD) (Nivre et al. 2016, 2020; de Marneffe et al. 2021), which assume word forms as elementary units.

However, it is not necessary to treat words as the elementary syntactic units of dependency structures. In the theory of Tesnière (1959), dependency relations are assumed to hold between slightly more complex units called **nuclei**. Nuclei are defined as semantically independent units consisting of a content word together with its grammatical markers, regardless of whether the latter are realized as morphological inflection or as independent words. In practice, a nucleus will often correspond to a single word—as in the English verb *chased*, where tense is realized solely through morphological inflection—but it may also correspond to several words—as in the English verb group *has chased*, where tense is realized by morphological inflection in combination with

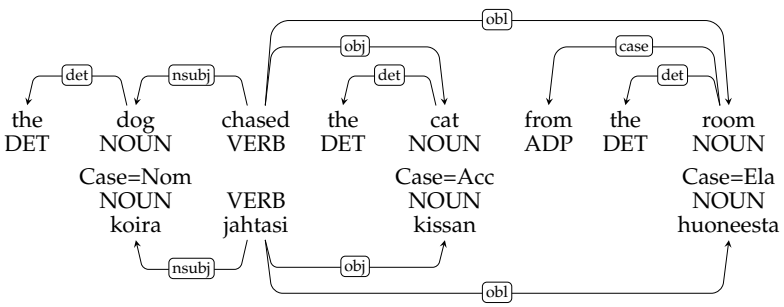


Figure 1
Word-based dependency trees for equivalent sentences from English (top) and Finnish (bottom).

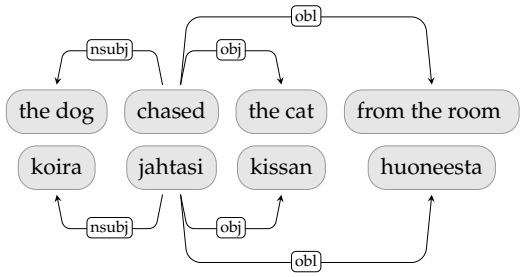


Figure 2
Nucleus-based dependency trees for equivalent sentences from English (top) and Finnish (bottom).

an auxiliary verb. A nucleus consisting of several words is known as a **dissociated nucleus**. It is easy to see that if we assume that the elementary syntactic units of a dependency tree are nuclei instead of words, then the English and Finnish sentences discussed above will be assigned identical dependency trees, visualized in Figure 2, and will differ only in the realization of the nuclei involved. Thus, whereas all nuclei in the Finnish sentence are simple nuclei, consisting of single words, all the nominal nuclei in English are dissociated nuclei, involving nouns together with standalone articles and the preposition *from*.

Can modern dependency parsers, based on neural network techniques and continuous word representations, benefit from a concept of nucleus inspired by the theory of Tesnière? This is a question that was first investigated in Basirat and Nivre (2021), where we proposed two key ideas. The first is the idea that we can define syntactic nuclei in UD representations, exploiting the fact that the UD guidelines prioritize dependency relations between content words that are the cores of syntactic nuclei, which makes it relatively straightforward to identify dissociated nuclei. In this way, we can gain access to annotated resources for training and evaluation of parsers across a wide range of languages. The second is the idea that transition-based parsers, as previously shown by de Lhoneux, Stymne, and Nivre (2020), can relatively easily be extended to include operations that create internal representations of syntactic nuclei. This gives us a vehicle for studying their impact on parsing performance. Basirat and Nivre (2021) includes a small experimental study, indicating that internal nucleus representations give small but consistent improvements in parsing accuracy.

In this article, we begin by replicating these experiments on a larger sample of languages, with a more detailed comparison of different models. We then try to analyze in more detail why nucleus composition only gives relatively modest accuracy improvements, which linguistic constructions benefit from these improvements, how we can explain the different rates of improvement across languages, and what information is encoded in the nucleus representations created through composition.

2. Related Work

Grammar and Annotation. Most dependency-based grammar formalisms and annotation frameworks discard the nucleus as the basic syntactic unit in favor of the word, but exceptions do exist. In the multi-stratal Functional Generative Description framework (Sgall, Hajičová, and Panevová 1986), nucleus-like concepts are captured at the tectogrammatical level, a property that is inherited by the three-layered annotation scheme of the Prague Dependency Treebank (Böhmová et al. 2003). Kahane (1997) introduces

the notion of a bubble tree to be able to represent verbal and nominal nuclei, as well as coordinate structures. Bārzdiņš et al. (2007) propose a syntactic analysis model for Latvian based on the x-word concept, which has clear affinities with the nucleus concept. In this approach, an x-word acts as a non-terminal symbol in a phrase structure grammar and can appear as a head or dependent in a dependency tree. Nespore et al. (2010) compare this model to the original dependency formalism of Tesnière (1959). Finally, Sangati and Mazza (2009) develop an algorithm to convert English phrase structure trees in Penn Treebank style to dependency trees that cover all of Tesnière's key concepts, including nuclei. More recently, we have shown in Basirat and Nivre (2021) that syntactic nuclei can be distinguished in UD treebanks, without the need for formal conversion.

Syntactic Parsing. Järvinen and Tapanainen (1998) propose Functional Dependency Parsing as an adaptation of Tesnière's dependency grammar for computational processing. They argue that the nucleus concept is crucial to establish cross-linguistically valid criteria for headedness and that it is not only a syntactic primitive but also the smallest semantic unit in a lexicographical description. While the approach of Järvinen and Tapanainen (1998) is based on linguistic rules, built on top of a constraint grammar system, Samuelsson (2000) instead defines a generative statistical model for nucleus-based dependency parsing, which however has never been implemented and tested. More broadly speaking, the nucleus concept has affinities with the chunk concept found in many approaches to parsing, starting with Abney (1991), who proposed to first find chunks and then dependencies between chunks, an idea that was generalized into cascaded parsing by Buchholz, Veenstra, and Daelemans (1999), among others. It is also clearly related to the vibhakti level in the Paninian computation grammar framework (Bharati and Sangal 1993; Bharati et al. 2009). In a similar vein, Kudo and Matsumoto (2002) use cascaded chunking for dependency parsing of Japanese with strictly head-final structures, a technique that was generalized to arbitrary (projective) dependency trees by Yamada and Matsumoto (2003). In a more recent study, de Lhoneux, Stymne, and Nivre (2020) investigate whether the hidden representations of a neural transition-based dependency parser encode information about syntactic nuclei, with special reference to auxiliary verb constructions. They find some evidence that this is the case, especially if the parser is equipped with a mechanism for recursive subtree composition of the type first proposed by Stenetorp (2013) and later developed by Dyer et al. (2015) and de Lhoneux, Ballesteros, and Nivre (2019). The idea is to use a composition operator that recursively combines information from subtrees connected by a dependency relation into a representation of the new larger subtree. In Basirat and Nivre (2021), we exploit this idea in combination with the UD-based definition of nuclei mentioned above, thus overcoming one of the major bottlenecks in earlier explorations of nucleus-based parsing, namely, the lack of annotated resources that can be coupled with an appropriate parsing model. This article is devoted to the further exploration of this approach.

3. Syntactic Nuclei in UD

Universal Dependencies (UD)[1] (Nivre et al. 2016, 2020; de Marneffe et al. 2021) is an open community effort aiming to provide cross-linguistically consistent morphosyn-

1 https://universaldependencies.org.

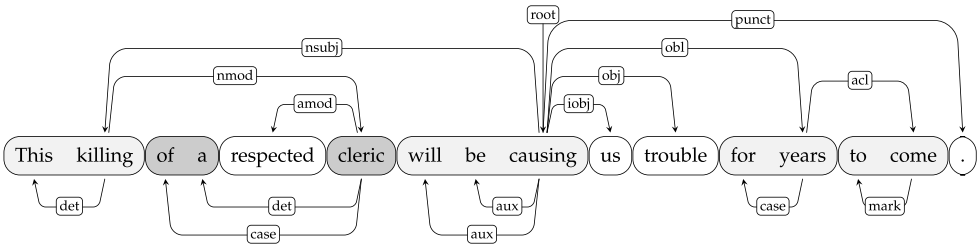


Figure 3
Syntactic UD representation with functional relations drawn below the sentence. Dissociated nuclei are grayed, with a darker shade for the discontiguous nucleus.

tactic annotation for as many languages as possible. The latest release from May 2022 (v2.10) features 228 annotated corpora, representing 130 languages from 22 language families. In this section, we review the proposal of Basirat and Nivre (2021) for how to define syntactic nuclei given the UD formalism.

The syntactic annotation in UD is based on dependency relations and the elementary syntactic units are assumed to be words, but the style of the annotation makes it relatively straightforward to identify substructures corresponding to (dissociated) nuclei. More precisely, UD prioritizes direct dependency relations between content words, as opposed to relations being mediated by function words, which has two consequences. First, incoming dependencies always go to the lexical core of a nucleus.[2] Second, function words are normally leaves of the dependency tree, attached to the lexical core with special dependency relations, which we refer to as functional relations.[3]

Figure 3 illustrates these properties of UD representations by showing the dependency tree for the English sentence *This killing of a respected cleric will be causing us trouble for years to come*. For perspicuity, functional relations are drawn below the sentence and other relations above it. Given this type of representation, we can define a *nucleus* as a subtree where all internal dependencies are functional relations, as indicated by the ovals in Figure 3. The nuclei can be divided into single-word nuclei (whitened) and dissociated nuclei (grayed). The latter can be contiguous or discontiguous, as shown by the nucleus *of a cleric*, which consists of the two parts colored with a darker shade.

This definition of nucleus in turn depends on what we define to be functional relations. For this study, we assume that the following 7 UD relations[4] belong to this class:

- Determiner (*det*): the relation between a determiner, mostly an article or demonstrative, and a noun. Especially for articles, there is considerable cross-linguistic variation. For example, definiteness is expressed by an independent function word in English (*the girl*), by a morphological inflection in Swedish (*flicka-n*), and not at all in Finnish.

2 Except in some cases of ellipsis, like *she did*, where the auxiliary verb *did* is "promoted" to form a nucleus on its own.
3 Again, there are a few well-defined exceptions to the rule that function words are leaves, including ellipsis, coordination, and fixed multiword expressions.
4 A more detailed description of the relations is available in the UD documentation at `https://universaldependencies.org`.

- Case marker (*case*): the relation between a noun and a case marker when it is a separate syntactic word and not an affix. UD takes a radical approach to adpositions and treats them all as case markers. Thus, in Figure 1, we see that the English adposition *from* corresponds to the Finnish elative case inflection.
- Classifier (*clf*): the relation between a classifier, a counting unit used for conceptual classification of nouns, and a noun. This relation is seen in languages that have a classification system, such as Chinese. For example, English *three students* corresponds to Chinese 三个学生, literally 'three [human-classifier] student'.
- Auxiliary (*aux*): the relation between an auxiliary verb or nonverbal marker of tense, aspect, mood, or evidentality and a verbal predicate. An example is the English verb group *will be causing* in Figure 3, which alternates with finite main verbs like *causes* and *caused*.
- Copula (*cop*): the relation between a verbal or nonverbal copula and a nonverbal predicate. For example, in English *Ivan is the best dancer*, the copula *is* links the predicate *the best dancer* to *Ivan*, but it has no counterpart in Russian *Ivan lucšij tancor*, literally 'Ivan best dancer'.
- Subordination marker (*mark*): the relation between a subordinator and the predicate of a subordinate clause. This is exemplified by the infinitive marker *to* in Figure 3. Other examples are subordinating conjunctions like *if*, *because*, and *that*, the function of which may be encoded morphologically or through word order in other languages.
- Coordinating conjunction (*cc*): the relation between a coordinator and a conjunct (typically the last one) in a coordination. Thus, in *apples, bananas, and oranges*, UD treats *and* as a dependent of *oranges*. This linking function may be missing or expressed morphologically in other languages.

The inclusion of the *cc* relation among the nucleus-internal relations is the most controversial decision, given that Tesnière treated coordination (including coordinating conjunctions) as a third type of grammatical relation—junction (fr. jonction)—distinct from both dependency relations and nucleus-internal relations. However, our goal in this paper is not to arrive at a faithful implementation of Tesnière's theory, but rather to explore how his concept of nucleus can be used as an inspiration in cross-linguistic investigations of dependency parsing. We therefore think coordinating conjunctions have enough in common with other function words to be included in this preliminary study and leave further division into finer categories for future work.[5]

Given the definition of nucleus in terms of functional UD relations, it would be straightforward to convert the UD representations to dependency trees where the elementary syntactic units are nuclei rather than words. However, as argued in Basirat and Nivre (2021), the usefulness of such a resource would currently be limited, given that it would require parsers that can deal with nucleus recognition, either in a preprocessing step or integrated with the construction of dependency trees, and such parsers are

5 In addition to separating the *cc* relation from the rest, such a division might include distinguishing nominal nucleus relations (*det*, *case*, and *clf*) from predicate nucleus relations (*aux*, *cop*, and *mark*).

not (yet) available. Moreover, evaluation results would not be comparable to previous research. We therefore exploit the nucleus concept in UD in two more indirect ways:[6]

- **Evaluation:** Even if a parser outputs a word-based dependency tree in UD format, we can evaluate its accuracy on nucleus-based parsing by simply not scoring the functional relations. This is equivalent to the Content Labeled Attachment Score (CLAS) previously proposed by Nivre and Fang (2017), and we will use this score as a complement to the standard Labeled Attachment Score (LAS) in our experiments.[7]
- **Nucleus Composition:** Given our definition of nucleus-internal relations, we can make parsers aware of the nucleus concept by differentiating the way they predict and represent dissociated nuclei and dependency structures, respectively. More precisely, we will make use of composition operations to create internal representations of (dissociated) nuclei, as discussed in detail in Section 4.

4. Syntactic Nuclei in Transition-Based Dependency Parsing

As explained in Basirat and Nivre (2021), the transition-based approach to dependency parsing (Yamada and Matsumoto 2003; Nivre 2003, 2004, 2008) is particularly well suited for integrating nucleus representations because of its incremental processing. A transition-based dependency parser constructs a dependency tree incrementally by applying transitions, or parsing actions, to configurations consisting of a stack S of partially processed words, a buffer B of remaining input words, and a set of dependency arcs A representing the partially constructed dependency tree. The process of parsing starts from an initial configuration and ends when the parser reaches a terminal configuration. The transitions between configurations are predicted by a history-based model that combines information from S, B, and A.

Like Basirat and Nivre (2021), we use a version of the arc-hybrid transition system initially proposed by Kuhlmann, Gómez-Rodríguez, and Satta (2011), where the initial configuration has all words $w_1, \ldots, w_n$ plus an artificial root node r in B, while S and A are empty.[8] There are four transitions: Shift, Left-Arc, Right-Arc, and Swap. Shift pushes the first word b_0 in B onto S (and is not permissible if $b_0 = r$). Left-Arc attaches the top word s_0 in S to b_0 and removes s_0 from S, while Right-Arc attaches s_0 to the next word s_1 in S and removes s_0 from S. Swap, finally, moves s_1 back to B in order to allow the construction of non-projective dependencies.[9]

Our implementation of this transition-based parsing model is based on the influential architecture of Kiperwasser and Goldberg (2016), which takes as input a sequence of vectors $x_1, \ldots, x_n$ representing the input words $w_1, \ldots, w_n$ and feeds these vectors

6 Basirat and Nivre (2021) in addition implemented an oracle parsing model by simulating perfect nucleus recognition. Because the insights gained from these experiments were limited, we refrain from replicating them in this article.

7 Our use of CLAS differs only in that we include punctuation in the evaluation, whereas Nivre and Fang (2017) excluded it.

8 Positioning the artificial root node at the end of the buffer is a modification of the original system by Kiperwasser and Goldberg (2016), inspired by the results reported in Ballesteros and Nivre (2013).

9 This extension of the arc-hybrid system was proposed by de Lhoneux, Stymne, and Nivre (2017), inspired by the corresponding extension of the arc-standard system by Nivre (2009).

through a BiLSTM that outputs contextualized word vectors $v_1, \ldots, v_n$, which are stored in the buffer B. Parsing is then performed by iteratively applying the transition predicted by a multilayer perceptron (MLP), taking as input a small number of contextualized word vectors from the stack S and the buffer B. More precisely, in the experiments reported in this article, the predictions are based on the two top items s_0 and s_1 in S and the first item b_0 in B. In a historical perspective, this may seem like an overly simplistic prediction model, but recent work has shown that more complex feature vectors are largely superfluous thanks to the BiLSTM encoder (Shi, Huang, and Lee 2017; Falenska and Kuhn 2019).

The baseline transition-based parser does not provide any mechanism for modeling the nucleus concept. It is a purely word-based model, where any more complex syntactic structure is represented internally by the contextualized vector of its head word. Specifically, when two substructures h and d are combined in a Left-Arc or Right-Arc transition, only the vector $\vec{h}$ representing the syntactic head is retained in S or B, while the vector $\vec{d}$ representing the syntactic dependent is removed from S. In order to make the parser sensitive to (dissociated) nuclei in its internal representations, Basirat and Nivre (2021) follow de Lhoneux, Ballesteros, and Nivre (2019) and augment the Right-Arc and Left-Arc actions with a composition operation. The idea is that, whenever the substructures h and d are combined with functional relation label l, the representation of the new nucleus is obtained by adding to the vector $\vec{h}$ the output of a learned function $g(\vec{h}, \vec{d}, \vec{l})$. We refer to $g(\vec{h}, \vec{d}, \vec{l})$ as the **composition vector**, and the addition of this vector to $\vec{h}$ can be understood as a way of modifying the head representation to reflect properties of the entire nucleus.

Of the different composition models explored by Basirat and Nivre (2021), we concentrate on the most successful one, where the composition vector $g(\vec{h}, \vec{d}, \vec{l})$ is the output of a (single-layered) perceptron with sigmoid activation applied to the concatenation of $\vec{h}$, $\vec{d}$ and $\vec{l}$:[10]

$$g(\vec{h}, \vec{d}, \vec{l}) = \sigma(W(\vec{h} \odot \vec{d} \odot \vec{l}) + b) \tag{1}$$

where $\odot$ is the vector concatenation operator.

We refer to the parsing model that applies composition only to syntactic nuclei as defined in the previous section as the *nucleus* composition model.[11] For the purpose of analysis, we also experiment with two additional models: the *non-nucleus* composition model, which applies composition to all constructions that are *not* nuclei, and the *generalized* composition model, which applies composition to *all* constructions. Using $f(h, d, l)$ for the representation of a construction produced by attaching d to h with relation label l and F for the set of functional relations, we define the models in (2–4).

Nucleus Composition.

$$f(h, d, l) = \begin{cases} \vec{h} + g(\vec{h}, \vec{d}, \vec{l}) & \text{if } l \in F \\ \vec{h} & \text{otherwise} \end{cases} \tag{2}$$

10 This operation was found to work as well as or better than a number of more complex functions in the experiments of Basirat and Nivre (2021).

11 This model was called *soft* composition in Basirat and Nivre (2021) for reasons that are no longer relevant.

Non-Nucleus Composition.

$$f(h,d,l) = \begin{cases} \vec{h} + g(\vec{h},\vec{d},\vec{l}) & \text{if } l \notin F \\ \vec{h} & \text{otherwise} \end{cases} \quad (3)$$

Generalized Composition.

$$f(h,d,l) = \vec{h} + g(\vec{h},\vec{d},\vec{l}) \quad (4)$$

For the purpose of analysis, we also explore variants of all three models where the BiLSTM encoder has been ablated, which means that the vectors representing words are the non-contextualized representations of the input words consisting of a randomly initialized word embedding concatenated with the output of a character-level BiLSTM.[12]

5. Experiments

In the previous sections, we have described the concept of a syntactic nucleus, discussed how nuclei can be identified in UD annotation, and shown how we can make transition-based parsers aware of them by using nucleus composition. In the following, we present a series of experiments designed to study the effect of nucleus composition across a diverse sample of languages and in relation to complementary models of composition. In this section, we describe the data selection, the experimental settings, and the main parsing results. In the following section, we analyze the results in further detail and test a number of hypotheses about the impact of nucleus composition across languages.

5.1 Data Selection

We select a sample of 20 treebanks from UD 2.8 (Zeman et al. 2021) with the goal of increasing the coverage of non-Indo-European languages compared to Basirat and Nivre (2021) and avoiding multiple languages from the same branch of a language family. Our initial selection criteria also included a minimum treebank size of 50,000 tokens. However, to make sure that Indo-European languages do not make up more than half the sample, we loosen the size constraint to include Vietnamese and Wolof treebanks with just over 40,000 tokens and include two languages from the Semitic branch of the Afro-Asiatic family: Arabic and Hebrew.

Table 1 gives an overview of the chosen treebanks in terms of family and genus, overall size, and frequency of the functional relations we are interested in. While *case* and *det* are on average the most frequent functional relations, their usage and frequency varies greatly across languages—from Japanese or Hindi, where on average about every fifth relation is a *case* relation, to Finnish, where *case* and *det* relations are the least frequent of the functional relations (ignoring the *clf* relation) and only about 2 of each occur in every 100 relations. In terms of functional relations in general, Italian and Greek emerge as the languages with the highest relative frequency, whereas functional relations are quite rare in Korean and Turkish with less than 10 percent of all relations belonging to that group.

12 For more information about the technical details of the baseline parser, see de Lhoneux et al. (2017).

Table 1
Selected treebanks, their family, genus, and size (in words) as well as relative frequencies of different types of functional relations and all functional relations combined.

Language	Treebank	Family	Genus	Size	aux	case	cc	clf	cop	det	mark	Func
Arabic	PADT	Afro-Asiatic	Semitic	242K	0.60	14.29	5.11	0.00	0.16	0.76	2.71	23.63
Armenian	ArmTDP	Indo-European	Armenian	52K	5.04	3.03	4.10	0.00	2.01	3.46	1.67	19.30
Basque	BDT	Basque	Basque	121K	8.54	1.56	3.85	0.00	2.02	2.50	0.18	18.65
Chinese	GSD	Sino-Tibetan	Chinese	121K	1.83	6.31	1.42	1.82	1.45	1.35	5.75	19.93
Finnish	TDT	Uralic-Finnic	Finnish	202K	3.26	1.48	4.13	0.00	2.72	1.72	1.95	15.27
Greek	GDT	Indo-European	Greek	62K	3.81	8.47	3.19	0.00	0.94	19.12	1.83	37.37
Hebrew	HTB	Afro-Asiatic	Semitic	116K	0.45	16.26	2.93	0.00	0.69	11.55	3.32	35.19
Hindi	HDTB	Indo-European	Indic	352K	6.41	19.27	1.87	0.00	1.00	2.05	4.11	34.70
Indonesian	GSD	Austronesian	Malayo-Sumbawan	121K	0.00	9.87	2.96	0.00	0.87	3.71	1.31	18.72
Irish	IDT	Indo-European	Celtic	116K	0.00	13.44	3.14	0.00	1.32	8.15	5.79	31.84
Italian	ISDT	Indo-European	Romance	278K	2.77	14.01	2.73	0.00	1.15	16.30	2.11	39.08
Japanese	GSD	Japanese	Japanese	194K	8.90	21.34	0.42	0.00	1.26	0.49	4.06	36.47
Korean	GSD	Korean	Korean	80K	0.08	2.03	0.28	0.00	0.13	3.83	0.46	6.81
Latvian	LVTB	Indo-European	Baltic	252K	1.26	4.68	4.01	0.00	1.39	2.63	1.91	15.87
Persian	PerDT	Indo-European	Iranian	494K	2.73	14.17	4.24	0.00	1.27	2.05	2.39	26.85
Russian	Taiga	Indo-European	Slavic	197K	0.30	8.56	4.12	0.00	0.41	2.49	1.63	17.51
Swedish	Talbanken	Indo-European	Germanic	97K	2.65	10.02	3.70	0.00	1.77	5.08	4.01	27.23
Turkish	Kenet	Turkic	Southwestern	179K	0.49	2.11	1.68	0.01	0.00	4.33	0.35	8.97
Vietnamese	VTB	Austro-Asiatic	Viet-Muong	44K	1.34	5.35	3.80	0.00	0.95	3.60	0.49	15.52
Wolof	WTB	Niger-Congo	Northern-Atlantic	43K	7.46	5.46	3.09	0.00	1.36	7.09	4.14	28.59
Average				168K	2.90	9.08	3.04	0.09	1.14	5.11	2.51	23.88

Two observations concerning the treebank statistics are relevant for the analysis later on. The first observation concerns Japanese and Korean, which are structurally quite similar languages but nevertheless exhibit very different statistics concerning functional relations, especially the *aux* and *case* relations. The explanation is that the developers of the respective treebanks have opted for different approaches to word segmentation, as discussed in Han et al. (2020). The Japanese GSD treebank adopts the Short Unit Word standard for Japanese, which is essentially a morpheme-level segmentation. By contrast, the Korean GSD treebank essentially relies on whitespace for segmentation, meaning that segments correspond to *eojeol* units, which consist of content words together with some function words. As a result, the frequency of (some) functional relations is likely to be over-estimated in Japanese and under-estimated in Korean. The second observation concerns the *clf* relation, which is found as expected in the Chinese GSD treebank but is conspicuously absent in the Vietnamese VTB treebank, despite the fact that Vietnamese is one of the prototypical classifier languages. We hypothesize that this is the result of an imperfect annotation conversion. In addition, there are 19 unexpected instances of the *clf* relation in the Turkish Kenet treebank, which we assume to be annotation errors. Given that the *clf* relation only (reliably) occurs in one of the 20 languages in our sample, we will omit it from some of the subsequent analysis.

The texts in the different treebanks are not parallel and come from a range of different genres and domains. The most common genre is newstext, which is found in the treebanks for all languages except Turkish and Wolof, and five treebanks consist solely of newstext (Arabic, Basque, Hebrew, Hindi, and Vietnamese). Other common genres are Wikipedia and blogs, found in six treebanks each. The lack of homogeneity with respect to genre and domain is a potential confound for our analysis, given that these factors have an impact on syntactic complexity as well as lexical variation. However, because information about genre is only available at the treebank level, it has not been possible to take these factors into account in the analysis.

5.2 Experimental Settings

Following the approach in Basirat and Nivre (2021), we use UUParser (de Lhoneux et al. 2017; Smith et al. 2018), a greedy transition-based dependency parser based on Kiperwasser and Goldberg (2016) and the extended arc-hybrid transition system by de Lhoneux, Stymne, and Nivre (2017). The parser predicts transitions between parser configurations with an MLP with a single hidden layer. The configurations consist of vectors for the two items on top of the stack S and the first item in the buffer B. As previously stated, in the baseline parser that does not apply composition, these vectors are simply contextualized word representations obtained by the BiLSTM over word and character vectors.

We refrain from using any kind of pre-trained representations, neither pre-trained word embeddings nor a full pre-trained language model that can be fine-tuned for the task. It is clear that such representations would improve parsing accuracy across the board, but our goal in this article is not to improve the state of the art but to analyze the impact of nucleus composition across typologically different languages. From this perspective, using pre-trained models would add a potential confound for the analysis, since the quality of available pre-trained models, whether monolingual or multilingual, varies considerably across languages. For the same reason, we stick to the well-studied architecture using a BiLSTM encoder, rather than a Transformer-based one, since the available evidence indicates that the performance difference is negligible when pre-trained representations are not used (Mohammadshahi and Henderson 2020).

Table 2 details the hyperparameter settings adopted in our experiments. In addition to parsers trained on these BiLSTM-generated features, we also train their counterparts without these features, instead using only the input word representations consisting of a concatenation of word and character embeddings trained along with the parser. This is to observe to what extent nucleus composition improves parsing accuracy when the parser is not supplied with any other contextual information to begin with.

Table 2
Hyperparameter settings for the parsing experiments.

Input representations	
Word embedding dimensions	100
Character embedding dimensions	100
Character BiLSTM output dimensions	100
BiLSTM encoder	
BiLSTM layers	2
BiLSTM dimensions (hidden/output)	125/125
Composition	
Label vector dimensions	10
MLP for transition prediction	
MLP hidden layers	1
MLP hidden dimensions	100
Dropout parameters	
Word dropout	0.33
Character dropout	0.33

Table 3
Parsing accuracy (LAS, CLAS) on 20 languages for 4 models (Base = baseline, NC = nucleus composition, NNC = non-nucleus composition, GC = generalized composition). Statistical significance for comparison of composition models to the baseline: $^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$.

	LAS				CLAS			
Language	Base	NC	NNC	GC	Base	NC	NNC	GC
Arabic	78.08	78.61**	78.56*	78.57	74.36	74.96**	74.96**	74.96
Armenian	74.85	75.44	75.69*	76.09**	71.73	72.47	72.64**	73.19**
Basque	73.58	74.21***	74.42*	74.89***	71.00	71.63***	72.06***	72.55***
Chinese	70.27	70.93**	71.59**	71.31**	68.43	69.02***	69.69***	69.41**
Finnish	79.00	79.52**	80.23**	80.13	78.06	78.56**	79.32***	79.17
Greek	83.19	84.08**	84.08*	83.96*	76.78	78.11***	78.07*	77.88**
Hebrew	82.80	83.29	83.18	83.26	76.14	76.95	76.82	76.97
Hindi	87.86	88.54	88.35	89.21***	83.83	84.68**	84.44	85.57***
Indonesian	76.51	76.98***	76.87*	76.73	74.35	74.95***	74.82	74.76
Irish	78.03	78.44	78.29*	78.24	71.83	72.47	72.24**	72.23
Italian	87.59	87.98	87.85	88.02	81.64	82.27	82.18	82.38
Japanese	92.91	92.93	92.96	92.92	90.17	90.21	90.25	90.30
Korean	74.98	75.35	75.68	75.86	74.75	75.20	75.58	75.70
Latvian	79.60	80.16**	80.52***	80.44***	78.46	79.01**	79.43***	79.33***
Persian	85.78	85.94	86.65***	86.88***	82.62	82.76	83.68***	83.97***
Russian	65.55	65.89	66.19	66.20**	60.94	61.20	61.58	61.60**
Swedish	77.05	77.77**	77.78*	78.15***	73.24	74.16*	74.26*	74.60**
Turkish	70.39	70.79	70.26***	70.28***	69.44	69.87	69.42***	69.41***
Vietnamese	56.95	57.56	58.17*	58.60**	54.74	55.69*	56.31***	56.67***
Wolof	73.17	74.21***	74.17**	74.35***	68.14	69.57***	69.49***	69.76***
Average	77.41	77.93	78.07	78.20	74.34	74.96	75.14	75.30

Models are trained for 50 epochs on the training sets of the selected 20 treebanks and evaluated on the respective development sets. For each combination of model type and treebank, we train 5 model instances with different random seeds and report the average score of the 5 runs. Statistical significance testing is performed using paired bootstrap permutation tests with 100,000 samples over all the 5 runs of the models under comparison.

5.3 Experimental Results

Table 3 shows the parsing accuracy of the baseline (Base) and nucleus composition (NC) parsers, as well as our two comparison composition models (NNC, GC) on 20 languages. It also shows which composition models obtain results that are significantly better than the baseline according to the bootstrap permutation test, with $* = p < 0.05$, $** = p < 0.01$, and $*** = p < 0.001$. Starting with the standard LAS score, we generally observe small improvements of up to 1 LAS point over the baseline parser when nucleus composition is used, but these improvements are only significant for 9 out of 20 languages: Arabic, Basque, Chinese, Finnish, Greek, Indonesian, Latvian, Swedish, and Wolof.[13] The greatest improvements are seen for Wolof, Greek, and Swedish.

13 It is worth noting that we use a different significance test than Basirat and Nivre (2021), who used a *t*-test for a difference in means based on the overall scores of the 5 runs. We believe that the test used here is more in line with standard practice in the parsing literature.

Non-nucleus and generalized composition also generally improve over the baseline with the exception of Turkish. Here the differences are significant for 14 and 12 out of 20 languages, respectively. With non-nucleus composition, the largest improvements are seen for Basque, Chinese, and Vietnamese, while generalized composition produces the greatest improvements for Vietnamese, Hindi, and Basque.

Given that the focus of our investigation is on nucleus-aware parsing, we consider the CLAS scores more important, since they report the accuracy with which a parser predicts dependency structures with nuclei as the elementary syntactic units. Although the overall trends are quite similar to the LAS results, we see that the average improvement of nucleus composition over the baseline is slightly larger (0.65 vs. 0.52), and that the improvement is significant for a larger number of languages (11 vs. 9), with Hindi and Vietnamese as new languages. The improvements are quantitatively larger also for the generalized and non-nucleus models, but for the latter model the improvement is no longer significant for Indonesian. When comparing the three composition models, we see that both generalized and non-nucleus composition on average give slightly larger improvements over the baseline than nucleus composition. However, there are a number of languages for which nucleus composition gives better results. Thus, the improvement of nucleus composition is greater than the improvement of both the other models for Greek, Indonesian, Irish, and Turkish, and additionally better than the improvement with non-nucleus composition for Hebrew, Hindi, Italian, and Wolof. For Arabic, finally, all three models give very similar results.

6. Analysis

The experimental results reported in the previous section corroborates the findings of Basirat and Nivre (2021) on a larger sample of languages. Nucleus composition gives small but quite consistent improvements in parsing accuracy for a diverse sample of languages, although the difference to the baseline is not always large enough to reach statistical significance. We will now proceed to a deeper analysis of the results and of the characteristics of nucleus composition. The analysis will address the following research questions:

1. Why does (nucleus) composition only give modest improvements?
2. How effective is nucleus composition compared to composition in general?
3. Which linguistic constructions benefit most from nucleus composition?
4. Why is nucleus composition more effective in some languages than others?
5. What information is captured in the learned composition operation?

6.1 Composition and Contextual Embeddings

The general usefulness of composition functions in neural dependency parsing has been discussed in several previous papers. While the use of composition functions was claimed to be crucial for obtaining high accuracy in the Stack-LSTM parser proposed

by Dyer et al. (2015), later studies have found the positive effect to be marginal at best. de Lhoneux, Ballesteros, and Nivre (2019) provide an in-depth study of composition in combination with different LSTM-based encoders and conclude that most of the information that composition is meant to propagate is already captured by the contextual embedding of the head word produced by a BiLSTM encoder. In another study, de Lhoneux, Stymne, and Nivre (2020) found that, whereas composition generally does not improve overall parsing accuracy, it may be needed to capture certain properties of auxiliary-verb constructions, a specific type of syntactic nucleus. Against this background, it is reasonable to assume that the limited improvement achieved through nucleus composition is due to the fact that important information about the nuclei is already contained in the BiLSTM representations of the lexical cores. An indirect test of this hypothesis can be performed by comparing parsing models where the BiLSTM encoder has been ablated, so that composition instead applies (recursively) to non-contextualized word representations.

Table 4 shows experimental results for parsers without the BiLSTM encoder. As expected, all scores are considerably lower than the corresponding scores in Table 3, but we also see that the composition models improve substantially over the baseline, with statistically significant improvements for all languages. While the baseline scores are about 20 (LAS) and 25 (CLAS) points below the corresponding parser with BiLSTM representations, the generalized composition model reduces the gap to just under 10 points in both cases. As was the case for the BiLSTM parsers, generalized composition

Table 4
Parsing accuracy (LAS, CLAS) on 20 languages for 4 models without BiLSTM features (Base = baseline, NC = nucleus composition, NNC = non-nucleus composition, GC = generalized composition). Statistical significance for comparison of composition models to the baseline: $^{***}p < 0.001$.

	LAS				CLAS			
Language	Base	NC	NNC	GC	Base	NC	NNC	GC
Arabic	64.35	71.00***	69.47***	74.75***	57.81	66.07***	63.93***	70.54***
Armenian	52.28	58.37***	59.29***	63.40***	46.47	53.54***	54.31***	59.27***
Basque	51.24	57.05***	56.38***	61.10***	46.60	53.28***	52.50***	57.96***
Chinese	42.81	47.65***	49.80***	55.14***	39.46	44.77***	47.25***	53.27***
Finnish	55.81	63.69***	62.54***	69.29***	53.71	62.37***	61.12***	68.68***
Greek	63.82	73.30***	68.38***	76.82***	51.15	65.22***	58.11***	70.97***
Hebrew	61.73	73.56***	65.38***	77.10***	46.99	63.80***	52.11***	69.04***
Hindi	62.82	74.32***	68.80***	77.34***	50.70	66.48***	58.68***	70.47***
Indonesian	58.06	66.57***	65.70***	73.51***	52.51	62.82***	61.70***	71.21***
Irish	62.73	70.76***	66.95***	73.95***	51.44	62.53***	57.42***	66.96***
Italian	66.68	77.91***	71.39***	82.20***	51.38	68.36***	58.53***	75.10***
Japanese	61.38	75.32***	65.93***	78.87***	44.18	64.25***	50.35***	69.29***
Korean	53.96	55.42***	65.26***	67.19***	54.10	55.61***	65.11***	67.01***
Latvian	57.08	62.76***	64.78***	69.85***	54.83	61.06***	63.23***	68.94***
Persian	62.40	75.06***	68.33***	78.24***	54.65	70.55***	61.73***	74.40***
Russian	49.02	53.89***	53.85***	58.36***	43.39	48.90***	48.50***	53.63***
Swedish	51.75	61.73***	58.78***	67.53***	44.88	56.96***	53.07***	64.10***
Turkish	59.94	62.39***	63.85***	66.10***	58.85	61.20***	62.79***	65.10***
Vietnamese	44.21	47.81***	48.40***	52.70***	40.86	45.04***	45.80***	50.49***
Wolof	53.58	62.34***	59.69***	67.71***	44.40	55.27***	52.26***	62.07***
Average	56.78	64.55	62.65	69.56	49.42	59.40	56.43	65.43

shows the largest average improvement, but this time generalized composition also performs best for each individual language. However, for the two other models, the trend is reversed in that nucleus composition gives a larger average improvement than non-nucleus composition with respect to both CLAS and LAS. Exceptions to this average trend are Armenian, Chinese, Korean, Latvian, Turkish, and Vietnamese both in LAS and CLAS, with Korean exhibiting the largest gap, with 9.8 LAS points and 9.5 CLAS points. Interestingly, Japanese emerges as the language with the largest overall improvement for both nucleus composition and generalized composition, which supports our earlier supposition that differences in word segmentation principles lead to an over-use of nucleus composition in Japanese and a corresponding under-use in Korean.

The experimental results considered in this section give rise to two observations. The first is the one already made by de Lhoneux, Ballesteros, and Nivre (2019), namely, that the use of composition operations to combine information from different substructures of a construction is almost superfluous in a neural transition-based parser that uses contextual embeddings, because the representation of the head is very likely to incorporate information about the dependent. The second observation is that, in the absence of contextual embeddings, nucleus composition is generally more effective than non-nucleus composition, which indicates that the need to incorporate information about the dependent is especially important in the case of nucleus-internal relations. We now proceed to examine whether this is true also in the presence of contextual embeddings, even though the improvements are much smaller in this case.

6.2 Nucleus Composition and Non-Nucleus Composition

As we observed in Section 5.3, non-nucleus composition on average gives greater improvement than nucleus composition for models with BiLSTM encoders. One possible explanation is that nuclei as we represent them here do not bring any special benefits and that we simply see greater improvement the more often we compose a head with its dependent. If this were true, we would expect the improvement of a composition model to be proportional to the frequency with which composition is applied when parsing a given data set. If this proportionality does not hold, on the other hand, it would be an indication that there is a qualitative difference between nucleus composition and non-nucleus composition.

To investigate this, we consider the **improvement ratio** of the two models, defined as the ratio between absolute improvement and absolute composition frequency where absolute improvement is the difference in the absolute number of correctly predicted nucleus-external dependencies between a given model and the baseline, and where the composition frequency is the absolute frequency of composed relations for a given model.[14] If we assume frequency to be the deciding factor, then we should see a similar improvement ratio for nucleus composition and non-nucleus composition, or possibly even a higher ratio in the non-nucleus setting if recursive composition were to give an additional improvement beyond that of an individual composition operation.

Table 5 shows absolute improvement, composition frequency, and improvement ratio for the languages in our samples. We observe that, while non-nucleus composition leads to a larger absolute improvement in nucleus-external relations than nucleus composition in 12 of the 20 languages, nucleus composition has a higher improvement ratio

14 Absolute frequencies are in all cases computed as averages over the 5 runs of a given model.

Table 5
Absolute improvement on nucleus-external relations, composition frequency, and improvement ratio (in percent) for nucleus composition (NC) and non-nucleus composition (NNC).

	Absolute improvement		Composition frequency		Improvement ratio	
Language	NC	NNC	NC	NNC	NC	NNC
Arabic	139.4	140.4	6,791	23,448	2.05	0.60
Armenian	32.0	39.2	1,064	4,295	3.01	0.91
Basque	124.2	206.8	4,492	19,603	2.77	1.06
Chinese	60.0	128.2	2,449	10,214	2.45	1.26
Finnish	78.0	194.2	2,901	15,407	2.69	1.26
Greek	86.8	84.2	3,917	6,526	2.22	1.29
Hebrew	61.4	51.6	3,869	7,543	1.59	0.68
Hindi	196.6	139.8	12,165	23,052	1.62	0.61
Irish	44.2	28.0	3,076	6,924	1.44	0.40
Indonesian	61.2	48.4	2,312	10,162	2.65	0.48
Italian	46.4	39.4	4,528	7,380	1.02	0.53
Japanese	2.6	5.8	4,482	7,805	0.06	0.07
Korean	49.6	92.2	811	11,147	6.12	0.83
Latvian	136.2	242.4	4,409	24,900	3.09	0.97
Persian	26.4	195.6	6,755	18,392	0.39	1.06
Russian	22.0	53.0	1,865	8,231	1.18	0.64
Swedish	64.6	72.0	2,727	7,070	2.37	1.02
Turkish	69.2	−2.6	1,620	15,935	4.27	−0.02
Vietnamese	92.0	152.2	1,773	9,741	5.19	1.56
Wolof	105.8	100.2	2,885	7,409	3.67	1.35
Average	74.9	100.6	3,744.6	12,259.2	2.49	0.83

for all languages but Persian and Japanese. Moreover, the difference is substantial for many languages. For example, in Korean, just over 800 nucleus compositions result in an improvement on about 50 dependency relations, while over 11,000 non-nucleus compositions give an improvement on less than 100 relations. To test whether the observed improvement ratios are consistent with our null hypothesis—that the improvement ratio of non-nucleus composition is equal to that of nucleus composition—we run a paired t-test. This test is highly significant at $p < 0.0001$. This suggests that nucleus composition is more powerful than non-nucleus composition for most languages, and that the greater absolute improvement observed for non-nucleus composition for a subset of the languages (six to be exact) is simply due to the much higher frequency of non-nucleus composition. For the remainder of this section, we will focus our analysis on the nucleus composition model.

6.3 Nucleus Composition and Linguistic Constructions

Which linguistic constructions benefit most from nucleus composition, and to what extent does this vary across languages? To address these interrelated questions, we will begin in this section by breaking down the accuracy improvements by different dependency relations and groups of relations. In the next section, we will take a complementary perspective and see to what extent the different rates of improvement across languages can be explained by linguistic factors such as the frequency of different types of functional relations.

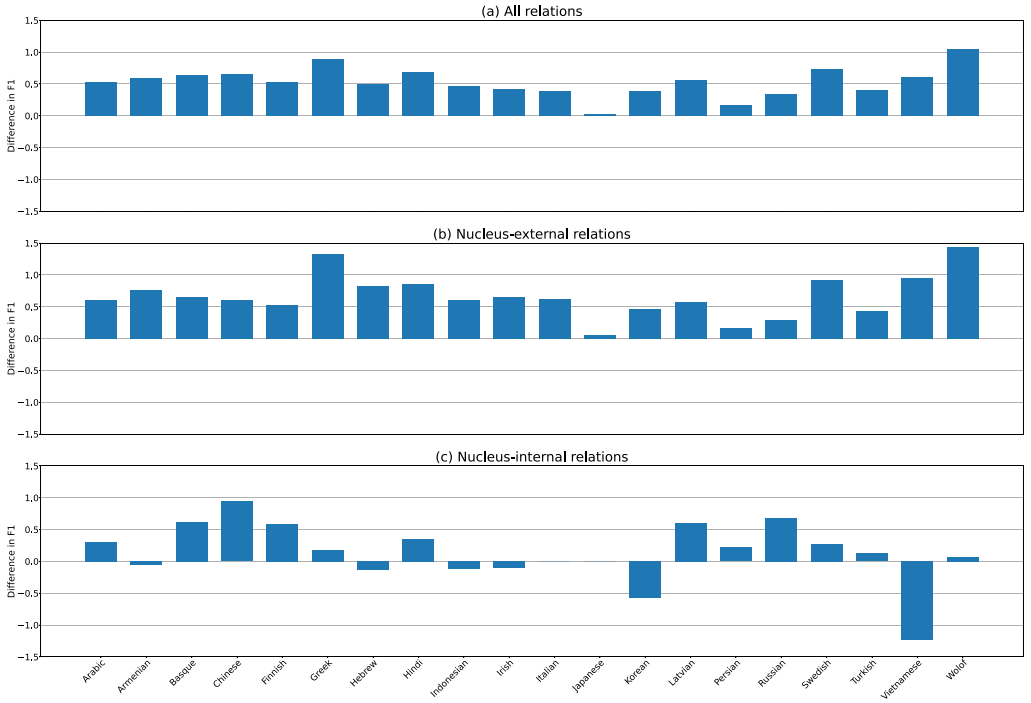


Figure 4
Absolute difference in labeled F1-score to the baseline for nucleus composition on different sets of relations.

Figure 4 illustrates the effect of nucleus composition on different relation types in different languages by breaking down the improvement over the baseline in labeled F1-score[15] for (a) all relations, (b) nucleus-external relations, and (c) nucleus-internal relations for each of the 20 languages. Just like Basirat and Nivre (2021), we observe that nucleus-external relation improvement is quite similar to overall relation improvement, indicating that the effect of composition is stronger in nucleus-external relations than in the internal relations. Exceptions to this generalization are the fairly strong improvement patterns in nucleus-internal relations for Chinese, Finnish, and Russian. In contrast to this, some languages like Indonesian, Korean, Turkish, and Vietnamese show a marked decrease in F1-score on nucleus-internal relations. However, due to the relatively low frequency of functional relations in these languages, even the more pronounced differences from the baseline only marginally affect the overall scores.

Zooming in on individual relations, and averaging across all languages, Figure 5 shows the improvement for each of the 37 universal syntactic relations, weighted by the relative frequency of the relation in order to show how much impact it makes on the overall improvement. We observe the *root* relation to be the one that shows the greatest improvement. This is expected, because it is the final relation that gets attached

15 Because the number of predicted relations of a certain type may be different from the number of true relations of that type, we have to use F1-score rather than accuracy.

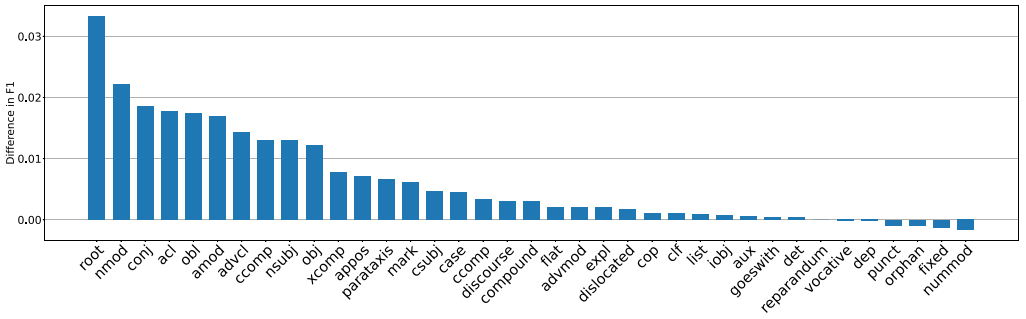


Figure 5
Difference in labeled F1-score of nucleus composition to the baseline for different UD relations weighted by the relative frequency of each relation and averaged across all languages.

and all previous correctly attached arcs should affect prediction of the root positively. In addition, composition of *mark* relations may help the parser distinguish main and subordinate clauses. The second largest improvement is found for *nmod*, which is the relation holding between a nominal modifier and its head nominal. One hypothesis for the improvement of *nmod*, together with other nominal dependents further down the list, such as *obl*, *obj*, and *nsubj*, is that composition of *case* relations may help the parser to better disambiguate different uses of nominals. The third most important relation is *conj*, which is the relation linking two coordinated constituents. In this case, it seems natural to assume that the composition of *cc* relations has a positive effect, since this may help the parser correctly choose the *conj* relation over the dependency relation assigned to the coordinated phrase as a whole. Among the relations that improve significantly we also find several relations that apply to clauses, such as *acl*, *advcl*, *ccomp*, and (to a lesser extent) *xcomp*, where the composition of *mark* relations can be hypothesized to have a positive effect. Finally, as was already visible in the breakdown by language in Figure 4 (c), we see that functional relations themselves tend to have lower relative improvement on average across all 20 languages.

Table 6 shows labeled F1-scores for the 10 relations improving the most in the 5 languages with the greatest overall improvement in terms of LAS. (Table 11 in Appendix A lists the top-10 relations for the full set of 20 languages.) We see that there is considerable variation in the top-10 lists, but also some stable patterns. Thus, both the *root* relation and the *conj* relation are in the lists for all languages except Chinese. By contrast, the *nmod* relation is high on the lists of Chinese and Hindi, but absent in Swedish and Wolof. Wolof is special also by having a very peaked distribution, with one relation (*root*) showing a much stronger improvement than all other relations. On the whole, however, nominal and clausal relations account for about half of the relations on all lists.

A tentative conclusion from the analysis in this section is that nucleus composition in general improves the identification of main predicates (*root*) and coordination relations (*conj*), as well as the disambiguation of nominals and subordinate clauses. For each of these cases, we can formulate specific hypotheses about which functional relations are involved in the improvement, and in the next section we will test whether properties of these relations can explain the differential improvement across languages.

Table 6
Improvement in labeled F1-score, weighted by relative frequency, for the 10 best UD relations in the 5 languages with greatest LAS improvements over the baseline (nucleus composition).

Wolof		Greek		Swedish		Hindi		Chinese	
relation	Δ F1	relation	Δ F1	relation	Δ F1	relation	Δ F1	relation	Δ F1
root	0.20	advcl	0.07	root	0.07	obj	0.06	nmod	0.06
advcl	0.11	nsubj	0.07	conj	0.06	nmod	0.04	acl	0.06
obl	0.11	root	0.05	amod	0.06	nsubj	0.04	advmod	0.06
acl	0.10	acl	0.03	acl	0.06	root	0.02	obj	0.05
ccomp	0.07	nmod	0.03	nsubj	0.03	aux	0.01	advcl	0.04
obj	0.07	obl	0.03	advmod	0.03	mark	0.01	obl	0.04
mark	0.05	conj	0.03	case	0.03	obl	0.01	nsubj	0.03
conj	0.04	appos	0.02	expl	0.02	conj	0.01	appos	0.03
dislocated	0.04	csubj	0.02	appos	0.02	xcomp	0.00	clf	0.02
advmod	0.03	obj	0.02	obj	0.02	amod	0.00	case	0.02

6.4 Nucleus Composition and Cross-Linguistic Variation

Thus far, we have considered general trends for overall LAS and CLAS, the effect composition has with and without contextual embeddings, the contrast between nucleus composition and non-nucleus composition, and the improvement in individual relations and groups of relations. In this section, we attempt to separate the language-specific patterns that affect parser improvement from more universal ones in order to understand how the individual functional relations affect the parser and its performance. Using linear mixed-effects models, we model CLAS[16] improvement over the baseline using different types of effects for each of the seven composed relations. The kinds of effects we consider are, for each functional relation l:

1. $H_{\text{POS}}(l)$ = entropy of the part-of-speech distribution for heads of l, abbreviated to *l POS entropy* in tables and running text
2. $H_{\text{REL}}(l)$ = entropy of the dependency label distribution for heads of l, abbreviated to *l rel entropy*
3. $dl(l)$ = average dependency length for l, abbreviated to *l dep length*
4. $rf(l)$ = relative frequency of l, abbreviated to *l frequency*

We choose these types of effects because they capture different aspects of the usage of a functional relation in a given language. For example, the two entropy effects for part of speech and dependency type of the head should give us an idea of how much uncertainty there is about the types of lexical core that the functional relation attaches to and the kind of functions the nucleus may fulfil. In cases where there is high entropy, especially for the head relation type, we would expect composition to be

16 We focus on CLAS, rather than LAS, because we are primarily interested in how nucleus composition affects nucleus-external relations.

useful if the function word encodes information about the relation type. For example, we hypothesized in the previous section that the *case* relation should be informative for the disambiguation of nominal functions. Another potentially significant factor is the (average) length of a functional relation dependency, assuming that contextual information about the functional relation decreases as the distance from the function word to its lexical core increases and that composition may compensate for this loss of information. Finally, the (relative) frequency of a functional relation directly translates to how often the functional relation is composed, which should correlate with CLAS improvement if the composition does improve parsing accuracy.

All features are calculated from the predicted trees for each of the five runs for a given language and mean normalized. We use linear mixed-effects (LME) models as implemented in the R package lme4 (Bates et al. 2015) with a random effect for language realized as a random intercept. Starting with a model made up only of the random effect for language, we gradually add fixed effects and perform likelihood ratio tests with the previous LME model to select those effects that significantly affect model likelihood. Because it is difficult to establish a single fixed hierarchy for the full set of 28 potential effects, and because the combination of effects and the order in which they are added affects which effects contribute significantly, we first identify significant effects for each of the four feature types individually. As part of this first step, we also train an additional LME model for both types of entropy effects, since they both express similar properties of the head. In a second step, we then try to fit a combined LME model including all the previously significant effects. Effects with missing values—for example, *clf* entropy and dependency length, which are not defined for the languages without *clf* relations—are omitted and models where we cannot fit the random effects because of singular fit issues are ignored.

With the individual LME models, we identify frequency of the *det* relation and POS entropy for the *cc* relation as significant effects. In the combined entropy model, both POS entropy and relation entropy for the *cc* relation emerge as significant. Combining these effects in a single model, all three remain significant and retain positive estimates. In other words, as the respective frequency and entropy increase, so does the improvement in CLAS. The resulting LME model is formulated as follows

$$\Delta_{CLAS} = rf(det) + H_{\text{REL}}(cc) + H_{\text{POS}}(cc) + (1|Language) \tag{5}$$

and a summary is displayed Table 7. The intercept here represents the micro-averaged improvement for all runs. We can see that all estimates are significantly different from zero and that *cc* POS entropy has the strongest effect. Seeing entropy of *cc* heads as a significant factor for CLAS improvement confirms that nucleus composition is of use when there is greater uncertainty about the head. Because *conj* relations are the most frequent incoming relation to heads of *cc*s, this should be related to the improved LAS on *conj* relations we observed in Section 6.3. Given our hypothesis that composed representations of dissociated nuclei are beneficial for dependency parsing, we would intuitively expect to see frequency effects of all functional relations to be significant in predicting improvement, as there should be a greater potential for parsing improvement, the more dissociated nuclei there are in a language. Instead, we see a significant effect only for *det* frequency, which is surprising given that *det* dependents normally do not encode information about the relation of a nominal to its head, at least not in the same obvious way as *case* dependents. The lack of frequency effects for other functional relations among the significant features may be an indicator that the information that

Table 7
LME model for improvement in CLAS due to nucleus composition.

Predictors	Estimates	CI	p
(Intercept)	0.65	0.56 – 0.76	<**0.001**
det frequency	0.59	0.20 – 0.98	**0.003**
cc rel entropy	0.77	0.27 – 1.26	**0.003**
cc POS entropy	0.79	0.30 – 1.28	**0.002**
Random Effects			
σ^2	0.17		
$\tau_{00\ language}$	0.01		
ICC	0.07		
$N_{language}$	20		
Observations	100		
Marginal R^2/Conditional R^2		0.266/0.315	

Table 8
LME model for functional composition improvement in CLAS in the setting without BiLSTM features.

Predictors	Estimates	CI	p
(Intercept)	9.99	9.31 – 10.66	<**0.001**
det frequency	6.06	3.28 – 8.84	<**0.001**
cop frequency	4.25	1.98 – 6.52	<**0.001**
aux frequency	3.83	1.49 – 6.17	**0.002**
case dep length	1.63	−0.34 – 3.60	0.104
case frequency	14.04	11.66 – 16.42	<**0.001**
Random Effects			
σ^2	0.27		
$\tau_{00\ language}$	2.28		
ICC	0.89		
$N_{language}$	20		
Observations	100		
Marginal R^2/Conditional R^2		0.900/0.989	

could be gained from composing them with a head is already present in the BiLSTM representations and thus does not contribute to improvement over the baseline. To investigate this, we apply the same strategy to the improvement of parsers without BiLSTM features and find that most of the frequency effects—excluding only *cc*, *mark*, and *clf*[17]—are significant here in addition to the length of *case* relations:

$$\Delta_{CLAS} = rf(det) + rf(cop) + rf(aux) + rf(case) + dl(case) + (1|Language) \quad (6)$$

The model summary can be found in Table 8, which shows that with the exception of *case* dependency length, for which the predicted coefficient is not significantly

17 We would not expect *clf* effects to be significant given that it only occurs in Chinese and very rarely in Turkish.

different from zero, all fixed effects are positively associated with improvement. In this setting, *case* frequency has the largest estimate. This model thus supports the theory we have for *det* being the only significant frequency feature in the model with contextual embeddings, namely that information about most of the other functional relations passed on by composition is already represented by the BiLSTM.

After this look at fixed effects and their estimates, we also consider the random effects sections of the model output. Here, σ^2 represents the residual variance and $\tau_{00\ \text{language}}$ is the between-group variance, which is the variance between the individual language intercepts and the average language intercept. ICC is the intra-class correlation coefficient—$\tau_{00\ \text{language}}$ divided by the total variance—and shows how much of the variance is explained by grouping the data by language. Marginal R^2 refers to the proportion of the total variance explained by the variance in fixed effects, whereas conditional R^2 considers the variance explained by both fixed and random effects. These two types of R^2 are defined as

$$R^2_m = \frac{\sigma^2_f}{\sigma^2_f + \sigma^2_\alpha + \sigma^2_\epsilon} \tag{7}$$

and

$$R^2_c = \frac{\sigma^2_f + \sigma^2_\alpha}{\sigma^2_f + \sigma^2_\alpha + \sigma^2_\epsilon}, \tag{8}$$

with σ^2_f referring to the variance of fixed effects, σ^2_α to that of random effects, and σ^2_ϵ to residual variance (Johnson and Schielzeth 2017).

Comparing models with and without BiLSTM features in this respect, we observe low values for residual variance, between-group variance, and a poor ICC of 0.07 for the model with BiLSTM features. This could reflect the fact that we predict improvement over the baseline rather than actual CLAS and the improvement values for the model with BiLSTM contextualized features are fairly small—specifically in the range of 0.04 to 1.42 CLAS—when the variance in a single language lies in a similar order of magnitude as that observed on all languages. This means that the language groups are fairly close to one another in terms of improvement, and individual random intercepts per group will be close to the average intercept. In contrast, the between-group variance in the LME model without BiLSTM is much higher, and the residual variance in proportion to it much lower, which means the ICC is quite strong at 0.89 (the average improvement of each of the 20 languages is within 1.51 and 20.07 CLAS).

As regards R^2, both marginal and conditional values are low in the BiLSTM setting, which indicates that the model does not explain a large part of the variation seen in the data and thus is not a very good fit, while the model without BiLSTM shows high R^2 values of 0.90 to 0.99 and consequently explains most of the variation observed in the data. This is in line with the more intuitive predictors of improvement we could identify for the setting without BiLSTM features.

To summarize the findings of this section, in the standard nucleus composition approach (with contextualized word embeddings), we identify the frequency of determiners as well as the entropy of coordinating conjunction heads in terms of both POS and head relation as effects contributing to CLAS improvement. The latter factors are plausible candidates to explain the improvement in *conj* dependencies, observed in the

previous section, and the frequency of determiners could be related to the improved disambiguation of nominal dependents. However, we do not find any clear correlates for the improved disambiguation of clausal dependents or for the increase in *root* accuracy.

6.5 Analysis of Composition Vectors

In nucleus composition, the composition vectors combine information from a *parent* vector, representing the core of the nucleus (either a single content word or a smaller nucleus in the recursive case), and a *child* vector, representing a function word. To gain insight into whether the composition process produces wholly distinct representations, or whether instead information from parent and/or child vectors is retained in the composed representations, we conduct a set of experiments using *diagnostic classifiers* (Hupkes, Veldhoen, and Zuidema 2018). Such an approach involves extracting intermediate representations generated by a neural model trained for a source task, and testing whether these representations have also encoded information relevant to a secondary target task. Diagnostic classifiers take the form of simple classifiers, trained independently of the source model, used as "probes" to monitor the degree to which secondary information is encoded as a by-product of training for the primary task. Examples of secondary tasks include part-of-speech tagging (Hewitt and Liang 2019), and testing whether verb agreement and transitivity is captured when training a syntactic parser (de Lhoneux, Stymne, and Nivre 2020).

The insight we aim to gather from these experiments is twofold—we first intend to measure *whether* any information from parent and/or child constituents is retained in their respective composed representations. Should this be the case, we also aim to measure the *degree* to which information from each of these constituents is retained, and whether there is any disparity in this retention. The specific probing task we use is prediction of the part of speech of the head word of the parent and child, respectively, and we train classifiers on the respective vectors of parent and child prior to composition, as well as their combined representation following composition, the composition vector. Belinkov (2018) previously found that while linear classifiers did not perform as well as single-layer nonlinear classifiers, trends across results for different kinds of input were preserved. As the purpose of this task is not to maximize tagging accuracy, but to measure differences in performance when using different vector representations, we therefore opt to use a simple linear SVM classifier.

The baselines we use in these probing experiments are the performance of the part-of-speech tagger on the input word representations learned by the parser during training, which we refer to as **input vectors**. These representations, which consist of the concatenation of a static word embedding and the output of a character-level BiLSTM, are well suited to this purpose as they provide a unique representation for each word type in each language, while containing no contextual information from the specific sentence. As can be seen in Table 9, there is a substantial difference in accuracy between the parent baseline at 72% and the child baseline at 92%. This difference can probably to a large extent be explained by a strong imbalance in the number of word types belonging to each category (that is, all languages have a greater number of content words than function words), but it is also possible that the learned input representations reflect grammatical categories to a higher degree for function words than for content words.

Our first set of probing experiments involves training the classifier to predict the part of speech of either parent or child, having trained on their respective vector

Table 9
Results of the linear SVM part-of-speech classifier trained on intermediate representations extracted from the nucleus composition model.

	Predict parent part of speech			Predict child part of speech		
Language	Baseline	Parent	Composition	Baseline	Child	Composition
Arabic	0.73	0.87	0.84	0.98	0.86	0.86
Armenian	0.69	0.69	0.78	0.92	0.65	0.84
Basque	0.73	0.71	0.73	0.94	0.76	0.76
Chinese	0.67	0.64	0.74	0.86	0.54	0.73
Finnish	0.58	0.81	0.78	0.96	0.78	0.78
Greek	0.75	0.79	0.87	0.98	0.83	0.81
Hebrew	0.76	0.80	0.80	0.96	0.77	0.77
Hindi	0.69	0.72	0.70	0.98	0.81	0.81
Indonesian	0.60	0.86	0.72	0.89	0.82	0.69
Irish	0.74	0.86	0.79	0.92	0.79	0.81
Italian	0.77	0.91	0.89	0.97	0.86	0.85
Japanese	0.82	0.85	0.85	0.93	0.92	0.92
Korean	0.76	0.84	0.85	0.82	0.81	0.84
Latvian	0.71	0.81	0.80	0.95	0.74	0.75
Persian	0.76	0.74	0.77	0.98	0.75	0.76
Russian	0.69	0.79	0.78	0.91	0.77	0.70
Swedish	0.74	0.80	0.76	0.91	0.65	0.76
Turkish	0.74	0.77	0.70	0.89	0.77	0.64
Vietnamese	0.69	0.71	0.77	0.84	0.63	0.68
Wolof	0.74	0.79	0.65	0.86	0.62	0.60
Average	**0.72**	**0.79**	**0.78**	**0.92**	**0.76**	**0.77**

representations (prior to composition), and predicting parent or child part of speech, having trained on the corresponding composition vectors. Table 9 contains the results of these preliminary probing experiments using vectors extracted from the nucleus composition model. Through these results, it is apparent that the classifier consistently outperforms the baseline in predicting the part of speech of the parent when trained on either the parent vectors or the composition vectors (with the exception of Chinese for both conditions, Persian for parent vectors, and Basque, Irish, and Korean for composition vectors). The same is not the case, however, for the performance of the classifier when predicting child part-of-speech tags, for which it uniformly underperforms when trained on the child and composed vector representations. Thus, while the BiLSTM encoder appears to facilitate disambiguation of parts of speech for content words, which is intuitively plausible, it appears to have a detrimental effect on the ability to assign syntactic categories to function words. At first, this result may seem surprising, but it is actually in line with some of the experimental results of de Lhoneux, Stymne, and Nivre (2020), according to which information about morphological agreement features of auxiliary verbs is better predicted from input word embeddings than from BiLSTM vectors. Taken together, these results seem to indicate that the contextualized representations of function words prioritize information about the sentential context at the expense of word-specific information.

We further note that training on composition vectors yields an increase over training on child vectors when predicting child part-of-speech tags, while the opposite is true when predicting parent part of speech. However, only the improved prediction of child part of speech is statistically significant according to a two-tailed t-test ($p = 0.015$). The question is what we can conclude from this trend. One thing to keep in mind is that the

Table 10
Average cosine distance for composition-parent vector pairs and composition-child vector pairs for the nucleus composition model. Results for all functional relations as well as per relation type.

	Composition-Parent							Composition-Child						
Language	All	aux	case	cc	cop	det	mark	All	aux	case	cc	cop	det	mark
Arabic	1.08	1.02	1.12	1.04	0.91	0.96	1.05	0.83	0.92	0.89	0.92	0.89	0.91	0.89
Armenian	1.02	1.06	1.05	1.04	1.00	1.01	1.04	0.87	1.01	0.94	0.98	1.01	1.00	1.02
Basque	1.10	1.16	1.06	1.06	1.08	1.04	0.92	0.84	0.94	0.88	0.93	0.9	0.92	0.89
Chinese	1.12	0.99	1.04	1.02	1.07	0.98	1.02	0.88	1.03	0.97	0.96	0.99	0.94	0.99
Finnish	1.08	1.07	1.09	1.08	1.07	1.03	1.10	0.89	1.05	0.98	0.97	1.03	1.01	1.02
Greek	1.07	1.08	1.09	1.05	0.97	0.98	1.11	0.85	1.00	1.02	0.98	0.96	0.95	0.96
Hebrew	1.14	1.05	1.15	1.13	1.02	1.12	1.19	0.89	0.92	1.01	0.99	0.99	0.98	0.97
Hindi	1.11	1.09	1.12	1.09	1.05	1.06	1.12	0.90	0.93	0.89	0.92	0.87	0.96	0.91
Indonesian	1.04	0.00	1.07	1.00	0.98	0.96	1.02	0.89	0.00	0.94	0.94	0.97	0.94	0.91
Irish	1.03	0.00	1.11	1.10	1.07	1.00	1.05	0.88	0.00	0.96	0.95	0.97	0.97	0.98
Italian	1.06	1.06	1.10	1.07	1.04	1.01	1.11	0.91	1.05	1.00	0.97	0.99	0.99	1.01
Japanese	1.02	0.99	1.04	1.08	1.03	1.06	1.00	0.90	0.89	0.86	0.89	0.86	0.96	0.93
Korean	1.01	1.02	1.02	1.01	1.02	1.00	1.10	0.86	0.96	0.96	0.99	0.90	0.95	0.97
Latvian	1.07	1.10	1.10	1.07	1.04	1.03	1.09	0.91	1.03	0.97	0.97	1.01	1.01	1.01
Persian	1.13	1.13	1.13	1.13	1.07	1.03	1.13	0.86	1.01	0.99	0.95	0.94	0.99	0.95
Russian	1.10	1.07	1.16	1.15	0.99	1.00	1.14	0.92	1.00	0.97	0.96	1.00	0.97	0.97
Swedish	1.04	1.16	1.11	1.10	1.05	1.03	1.14	0.86	1.03	0.93	0.95	0.97	0.95	0.99
Turkish	0.97	1.02	1.01	1.02	0.00	1.06	1.06	0.83	0.98	0.95	1.00	0.00	0.98	0.97
Vietnamese	1.01	0.95	1.00	0.98	0.94	0.93	0.93	0.89	0.96	1.02	0.99	0.99	0.99	0.85
Wolof	1.02	0.99	1.01	1.05	0.98	1.00	1.03	0.89	0.98	0.99	0.98	0.96	1.02	0.96
Average	1.05	1.06	1.08	1.06	1.02	1.01	1.07	0.96	0.98	0.96	0.96	0.96	0.97	0.96

vector used in the prediction of parser transitions is the composition vector added to the parent vector. It is therefore possible that the composition will emphasize information from the child vector more. To test this assumption, we take the average cosine distance between the parent vector and composition vector, and between the child vector and composition vector, for each functional relation.[18] The results can be found in Table 10, and the trend is clearly that child vectors have a lower cosine distance to the composition vector than the corresponding parent vector (that is, for a given functional relation, the child vector will generally be more similar to the composition vector than the parent vector). This trend holds for all languages and functional relations, with the exception of the *cop* relation for Hindi and Russian, and all relations except *mark* for Vietnamese. While this shows that composition vectors are generally more similar to child vectors than to parent vectors, this does not in itself explain why the composition vector gives (slightly) higher accuracy for predicting the child part of speech.

An additional approach we take to the analysis of composition vectors is to visualize them after dimensionality reduction. Our method involves extracting composition vectors generated by the model, as well as composition+parent and parent vectors, and using t-SNE (Van der Maaten and Hinton 2008) to reduce the dimensionality of each vector to 2 dimensions. These two-dimensional representations are then plotted in order to give a visually intuitive representation of similarities and differences between vectors for words occurring in different types of nuclei (as defined by different functional

18 We omit the *clf* relation, which only occurs in Chinese and marginally in Turkish.

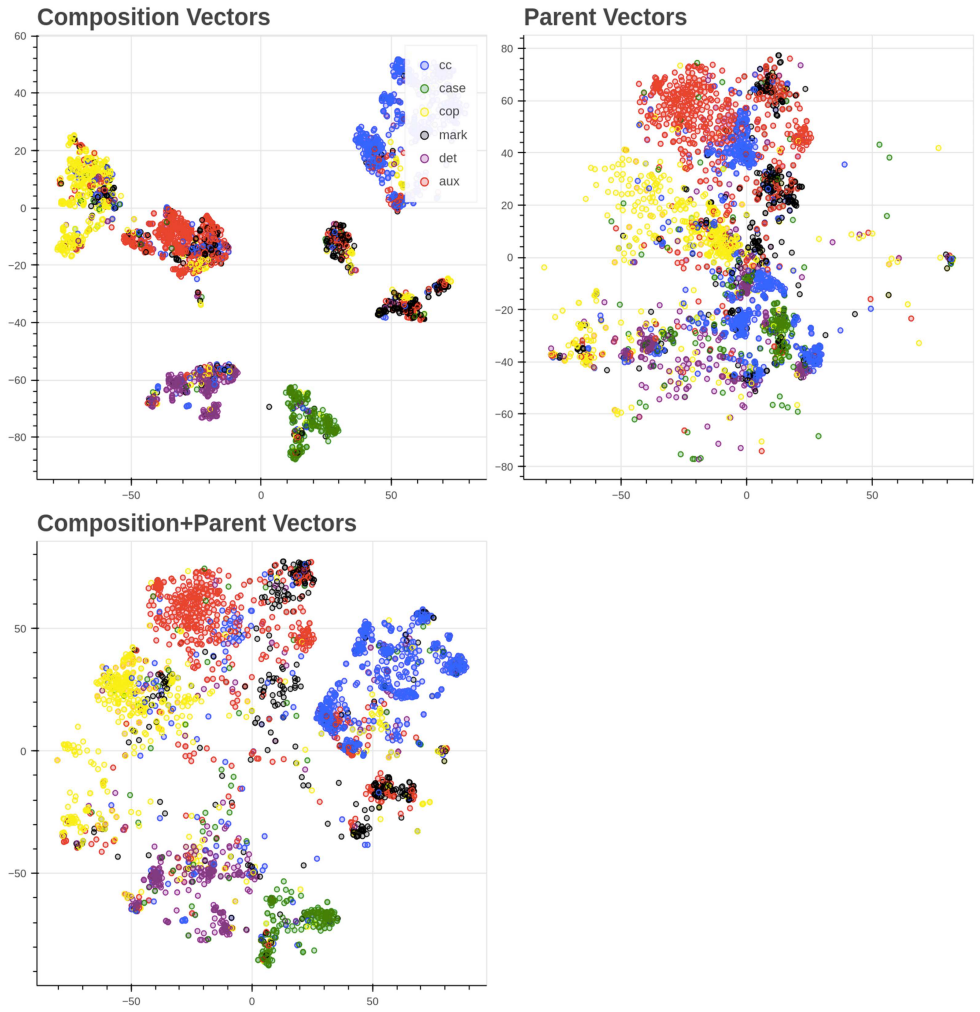


Figure 6
Visualization of different vector types for Finnish after dimensionality reduction to 2 dimensions, with color coding of different nucleus types.

relations). Such visualizations are especially informative with regard to illustrating the degree of separability between different types of nuclei, as such separations will be represented as tight and distinct clusters in the graph. Producing graphs for composition, composition+parent, and parent vectors allows us to compare how the cores of different types of nuclei are represented before and after composition.

Figure 6 contains the graphs for these three types of vectors for Finnish, with color coding of different nucleus types. The most striking pattern is that the composition vectors form clearly defined clusters, predominantly grouped by nucleus type, with one cluster each dominated by *aux*, *case*, *cc*, *cop*, and *det* nuclei, and with two smaller clusters dominated by *mark* nuclei. For the corresponding parent vectors, no distinct clusters are visible, although there is still a tendency that cores belonging to the same nucleus type group together. The effect of adding the composition vectors to the parent vectors, visible in the third graph, is thus to shift parent vectors to enhance discrimination

between nucleus types, which in turn appears to be beneficial for predicting further parsing transitions.

The tendency for composition vectors to form distinct clusters is visible for all languages, as can be seen in Appendix B, although the degree of separation of different nucleus types varies. To confirm this effect without having to rely solely on visual inspection, we carry out a linear discriminant analysis, which shows that the accuracy with which nucleus type can be distinguished increases from 76 to 79 percent when composition vectors are added to parent vectors. This difference is statistically significant according to a paired t-test ($p = 0.00016$).

To sum up the analysis in this section, it appears that composition vectors in general encode more information about the function word than about the lexical core of a nucleus, and that adding such vectors to the lexical core representations increases the similarity of representations belonging to the same nucleus type.

7. Discussion

Our main experimental results corroborate the findings of Basirat and Nivre (2021) and show that composed representations of syntactic nuclei, defined in terms of functional relations in UD, give small but consistent improvements in dependency parsing accuracy over a wide range of typologically diverse languages. In the subsequent analysis, we have tried to shed light on how nucleus composition interacts with other components of the parser, how improvements in parsing accuracy are related to different linguistic constructions, and what information is captured by the composition vectors. We will now discuss what broader conclusions can be drawn from the analysis.

The idea of using recursive composition to build representations of complex syntactic structures in dependency parsing was first proposed by Stenetorp (2013) and later developed by Dyer et al. (2015), who showed that it can be a powerful technique for improving the accuracy of a parser where input words are represented by static word embeddings. In this setup, recursive composition can be understood as the neural counterpart of the hierarchical feature templates that were important to achieve high parsing accuracy in non-neural transition-based dependency parsers (Nivre, Hall, and Nilsson 2006; Zhang and Nivre 2011). However, later studies have shown that the need for recursive composition greatly diminishes when parsers are equipped with BiLSTM or Transformer encoders, which compute contextualized representations of the input words (Shi, Huang, and Lee 2017; de Lhoneux, Ballesteros, and Nivre 2019; Falenska and Kuhn 2019; de Lhoneux, Stymne, and Nivre 2020). Even though these encoders only have access to the sequential structure of the input sentence, they seem to be capturing enough contextual information to compensate for the lack of recursion or hierarchical structure.

When composition is only applied to nuclei, the degree of recursion is limited and the words involved are often close to each other in the input sequence. Therefore, it is hardly surprising that composition is almost redundant given the contextualized word representations. However, because we do see significant improvements for many languages, it seems that the learned composition function for nucleus elements nevertheless captures some additional information that helps the parser. This hypothesis is further strengthened by the observation that nucleus composition has a significantly higher improvement ratio than non-nucleus composition, a result that holds for all languages except Japanese and Persian, which show little improvement overall.

If it is true that composition is most effective (in relation to frequency) when applied to nucleus-*internal* relations; it is also true that most of the improvement in parsing

accuracy results from better prediction of nucleus-*external* relations. Moreover, the relations that improve the most, both on average and for most languages, are central dependency relations involving nominal and clausal dependents, as well as coordination and identification of main clause predicates. This is consistent with Tesnière's original conception of syntactic structure as consisting of dependency relations holding between syntactic nuclei, and thus gives some support to the idea that nucleus representations can be beneficial for the analysis of this structure. The main difference between our implementation of this idea and Tesnière's own theory lies in the analysis of coordinating conjunctions as markers of coordination, which we have subsumed under the notion of nucleus, whereas Tesnière treats it as a category of its own.

Although the positive effect of nucleus composition is relatively consistent across languages, there is considerable variation both in the magnitude of the improvement and in the detailed analysis of which linguistic constructions benefit most from nucleus composition. Fully explaining this variation is a task well beyond the scope of this article, but by factoring out language-specific effects using a linear mixed-effects model, we have begun to identify some factors that appear to be stable and significant predictors of how much nucleus prediction will improve parsing accuracy for a given language.

Two of these factors are related to coordination and the *cc* relation, namely, the part-of-speech tag entropy and the relation entropy of the head of the *cc* relation, which both show a positive correlation with accuracy improvement. In other words, the harder it is to predict the tag and relation of the head, the more it helps to compose the head and the coordination marker. The third factor is the frequency of *det* relations, which also correlates positively with increased accuracy, suggesting that information encoded by determiners is useful for disambiguating the syntactic role of nominals. All of these factors make intuitive sense, but it is slightly surprising that these are the only nucleus-related factors that come out as significant. In particular, we had hypothesized that factors related to the *case* relation, and perhaps to a lesser extent the *mark* relation, would be important, given that case markers and subordinators do encode information about the syntactic role of the nominals or clauses they belong to.

Whether this lack of significance is due to a genuine lack of effect, or rather to the combination of a small numerical improvement and a large inter-language variation, is impossible to say with certainty. However, the results obtained for the model with no BiLSTM encoder suggests that it may be the latter. For that model, not only *det* frequency but also *case*, *aux*, and *cop* frequency are highly significant factors, which together explain over 90% of the variance. Interestingly, however, no factor related to the *cc* relation (nor to the *mark* relation) is significant in this case. To fully understand the mechanisms at play here, it seems we have to go into even more depth and analyze the patterns for individual languages, maybe even down to the sentence level, an investigation that we have to leave for future research. It is also worth remembering that the heterogeneity with respect to text genres in the different treebanks may play a role here (cf. Section 5.1).

Finally, when analyzing the effect of nucleus composition on the vector representations of nuclei and their components, we find that the learned composition operation tends to produce vectors that emphasize the function word part of a nucleus over its lexical core and creates distinct clusters that correlate with nucleus type. Adding these vectors to the lexical core representations in turn produces nucleus representations that cluster by type to a higher degree than the lexical core representations themselves. The capacity to capture nucleus types in this way is presumably what benefits parsing accuracy.

8. Conclusion

We have explored how Tesnière's concept of syntactic nucleus can be used to enrich the representations of a transition-based dependency parser, relying on UD treebanks for supervision and evaluation in experiments on a wide range of languages. We have corroborated previous experimental results showing that the use of composition operations for building internal representations of syntactic nuclei can lead to small but significant improvements in parsing accuracy for nucleus-external relations, notably in the analysis of coordination, nominal dependents, clausal dependents, and main predicates. We have presented evidence that nucleus composition is more effective than composition of other syntactic constituents, and we have shown that cross-linguistic variation can to some extent be explained by factors relating to entropy and frequency of function words. Finally, we have shown that the concrete effect of nucleus composition is to enhance the similarity of nucleus representations belonging to the same type.

Several lines of inquiry suggest themselves for future research. In order to gain a deeper understanding of the mechanisms by which nucleus composition improves parsing, it may first of all be worthwhile to study individual languages in more depth. Another idea would be to study the dynamics when processing individual sentences, for example, using causal analysis along the lines of Finlayson et al. (2021). Going beyond the parsing approach pursued in this article, it is an open question how nucleus representations can be integrated into parsing paradigms other than transition-based parsing. While traditional graph-based algorithms for dependency parsing assumes that the elementary syntactic units (normally words) are known prior to parsing, an assumption without which parsing would be computationally hard, recent neural incarnations of the graph-based paradigm do not rely on traditional dynamic programming or spanning tree algorithms and may therefore allow the integration of nucleus representations in the scoring model of the parser. Whether this would lead to improvements in parsing accuracy is of course a different question.

Appendix A: Top-10 Improved Relations in 20 Languages

Table 11
Improvement (or degradation) in labeled F1-score, weighted by relative frequency, for the 10 best UD relations for all 20 languages ordered by overall LAS improvement.

Wolof		Greek		Swedish		Hindi		Chinese	
relation	Δ F1	relation	Δ F1	relation	Δ F1	relation	Δ F1	relation	Δ F1
root	0.20	advcl	0.07	root	0.07	obj	0.06	nmod	0.06
advcl	0.11	nsubj	0.07	conj	0.06	nmod	0.04	acl	0.06
obl	0.11	root	0.05	amod	0.06	nsubj	0.04	advmod	0.06
acl	0.10	acl	0.03	acl	0.06	root	0.02	obj	0.05
ccomp	0.07	nmod	0.03	nsubj	0.03	aux	0.01	advcl	0.04
obj	0.07	obl	0.03	advmod	0.03	mark	0.01	obl	0.04
mark	0.05	conj	0.03	case	0.03	obl	0.01	nsubj	0.03
conj	0.04	appos	0.02	expl	0.02	conj	0.01	appos	0.03
dislocated	0.04	csubj	0.02	appos	0.02	xcomp	0.00	clf	0.02
advmod	0.03	obj	0.02	obj	0.02	amod	0.00	case	0.02

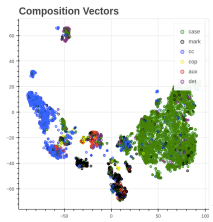
Basque		Vietnamese		Armenian		Latvian		Arabic	
relation	Δ F1	relation	Δ F1	relation	Δ F1	relation	Δ F1	relation	Δ F1
conj	0.07	xcomp	0.09	root	0.08	nmod	0.06	nmod	0.05
advcl	0.04	root	0.08	cc	0.04	root	0.04	obj	0.05
root	0.03	obj	0.04	nmod	0.04	obl	0.04	obl	0.03
xcomp	0.02	ccomp	0.04	xcomp	0.03	conj	0.03	conj	0.02
ccomp	0.02	amod	0.03	compound	0.03	nsubj	0.03	nsubj	0.02
nmod	0.02	parataxis	0.03	nsubj	0.03	obj	0.02	advmod	0.01
iobj	0.02	cc	0.02	discourse	0.02	xcomp	0.01	nummod	0.01
acl	0.02	nmod	0.01	csubj	0.02	amod	0.01	amod	0.01
advmod	0.01	advcl	0.01	amod	0.02	advmod	0.01	dep	0.01
obl	0.01	compound	0.01	aux	0.02	det	0.01	advcl	0.01

Finnish		Hebrew		Indonesian		Irish		Turkish	
relation	Δ F1	relation	Δ F1	relation	Δ F1	relation	Δ F1	relation	Δ F1
obl	0.07	nmod	0.02	ccomp	0.07	nmod	0.13	amod	0.05
conj	0.07	det	0.02	parataxis	0.05	obl	0.09	nsubj	0.04
acl	0.03	conj	0.02	advcl	0.05	root	0.03	csubj	0.03
obj	0.03	appos	0.02	nmod	0.04	nsubj	0.02	conj	0.03
amod	0.02	flat	0.01	xcomp	0.04	cc	0.02	advmod	0.02
ccomp	0.02	acl	0.01	compound	0.04	flat	0.01	case	0.02
root	0.02	ccomp	0.01	flat	0.03	list	0.01	obj	0.01
nsubj	0.01	root	0.01	obl	0.02	case	0.01	parataxis	0.01
cop	0.01	mark	0.01	det	0.02	advcl	0.01	discourse	0.01
nummod	0.01	amod	0.01	root	0.02	det	0.01	det	0.01

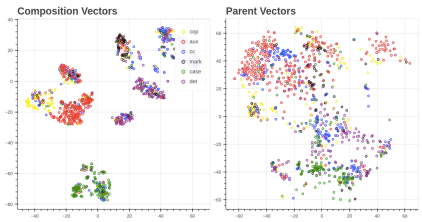
Italian		Korean		Russian		Persian		Japanese	
relation	Δ F1	relation	Δ F1	relation	Δ F1	relation	Δ F1	relation	Δ F1
conj	0.03	obj	0.07	amod	0.06	compound	0.02	nmod	0.02
ccomp	0.02	advmod	0.06	mark	0.04	nmod	0.02	nsubj	0.02
advcl	0.01	root	0.04	parataxis	0.04	obj	0.01	fixed	0.00
csubj	0.01	acl	0.03	flat	0.03	ccomp	0.01	csubj	0.00
appos	0.01	dep	0.02	iobj	0.02	acl	0.01	compound	0.00
xcomp	0.01	flat	0.02	case	0.02	advmod	0.01	dislocated	0.00
expl	0.01	appos	0.02	aux	0.02	conj	0.01	mark	0.00
iobj	0.01	amod	0.01	acl	0.01	mark	0.00	advmod	0.00
amod	0.01	nummod	0.01	discourse	0.01	det	0.00	nummod	0.00
acl	0.00	advcl	0.01	det	0.01	nsubj	0.00	obj	0.00

Appendix B: Vector Visualization after Dimensionality Reduction (t-SNE) for 20 Languages

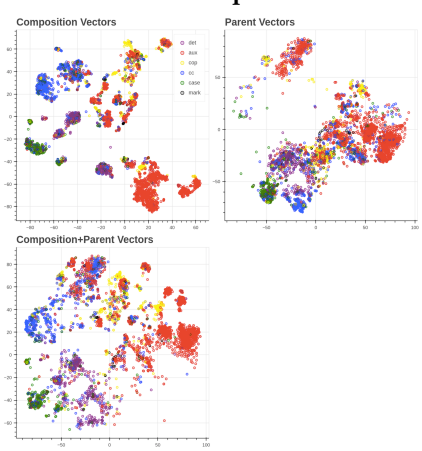
Arabic



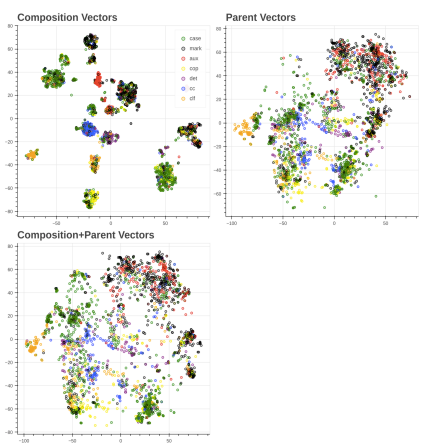
Armenian



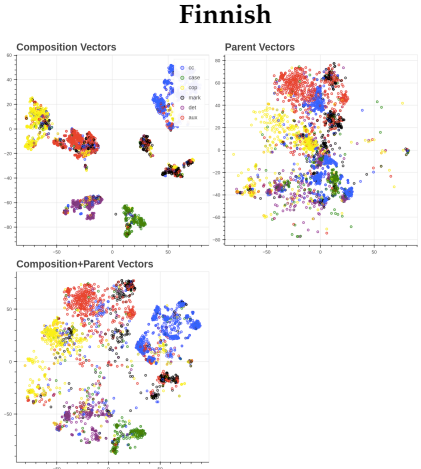
Basque



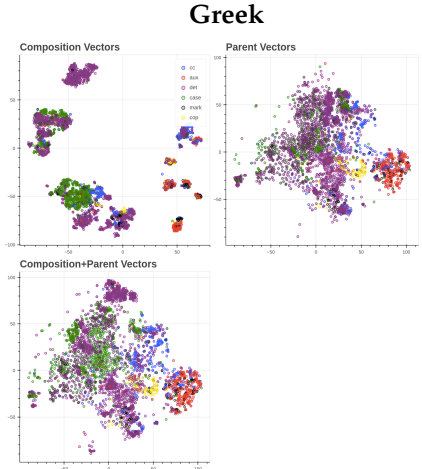
Chinese

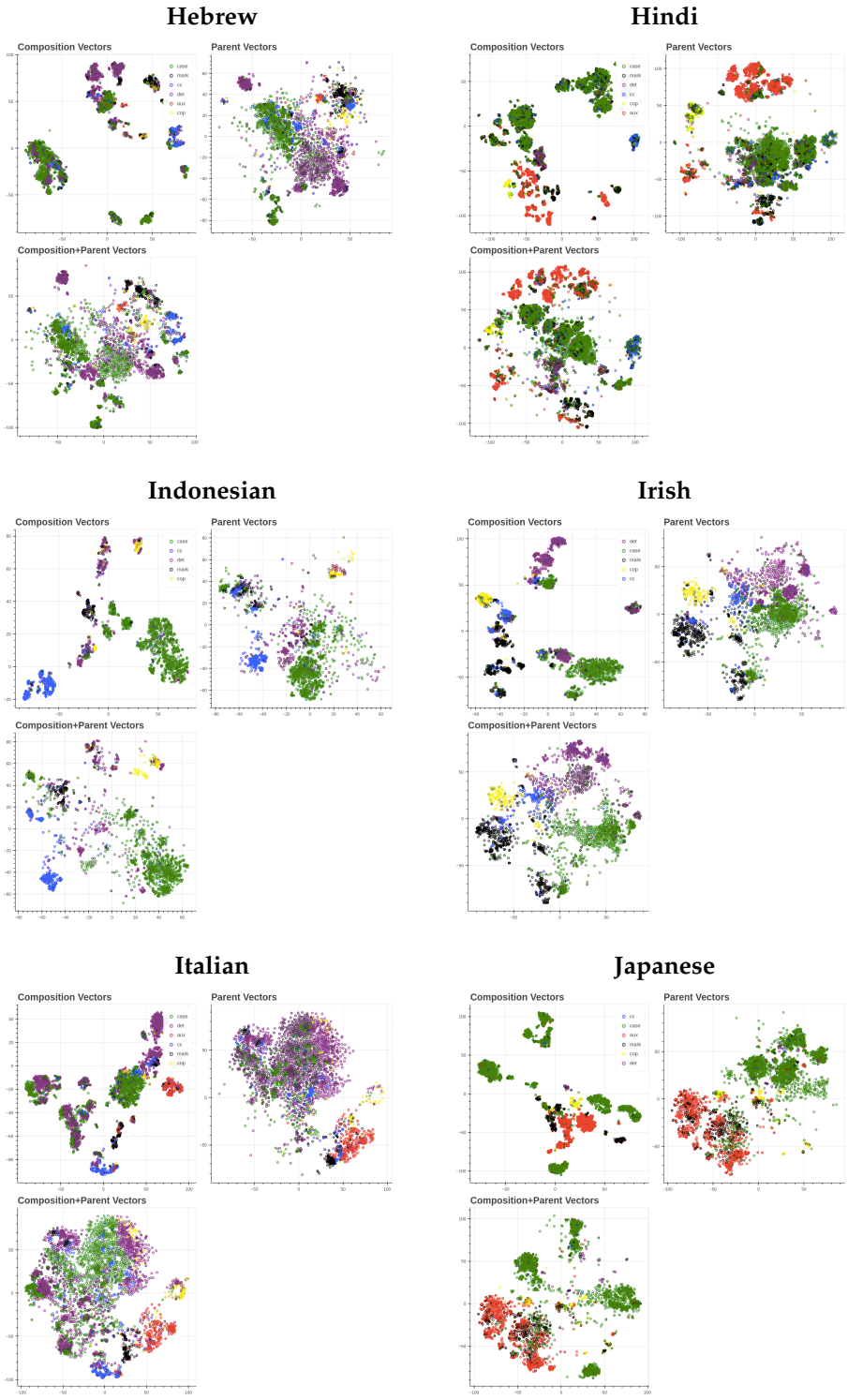


Finnish

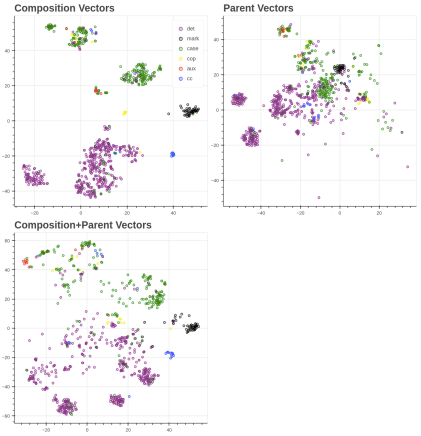


Greek

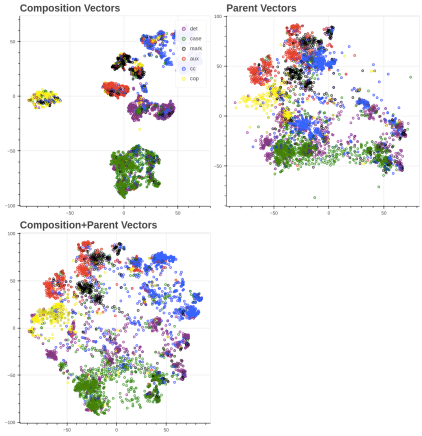



Hebrew
Composition Vectors
Parent Vectors
Composition+Parent Vectors
Hindi
Composition Vectors
Parent Vectors
Composition+Parent Vectors
Indonesian
Composition Vectors
Parent Vectors
Composition+Parent Vectors
Irish
Composition Vectors
Parent Vectors
Composition+Parent Vectors
Italian
Composition Vectors
Parent Vectors
Composition+Parent Vectors
Japanese
Composition Vectors
Parent Vectors
Composition+Parent Vectors

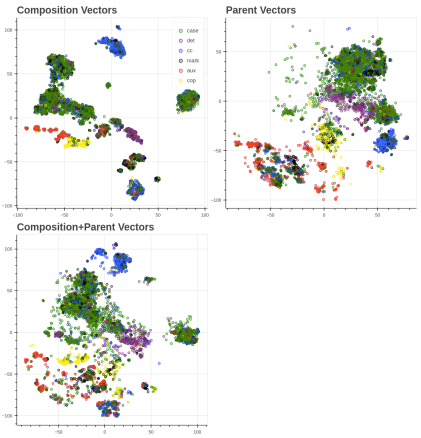
Korean



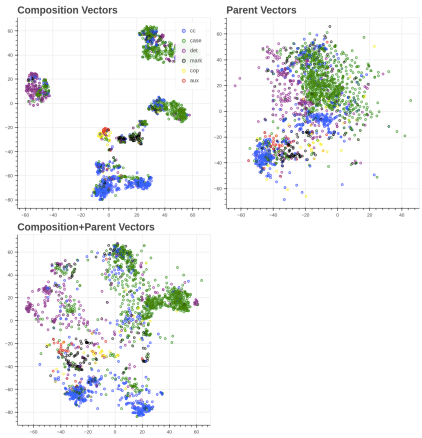
Latvian



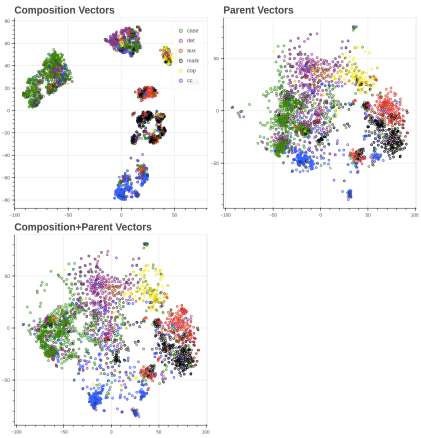
Persian



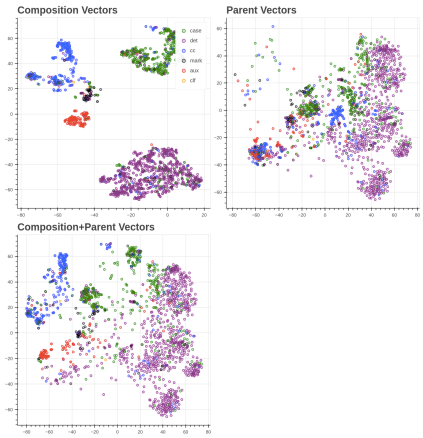
Russian

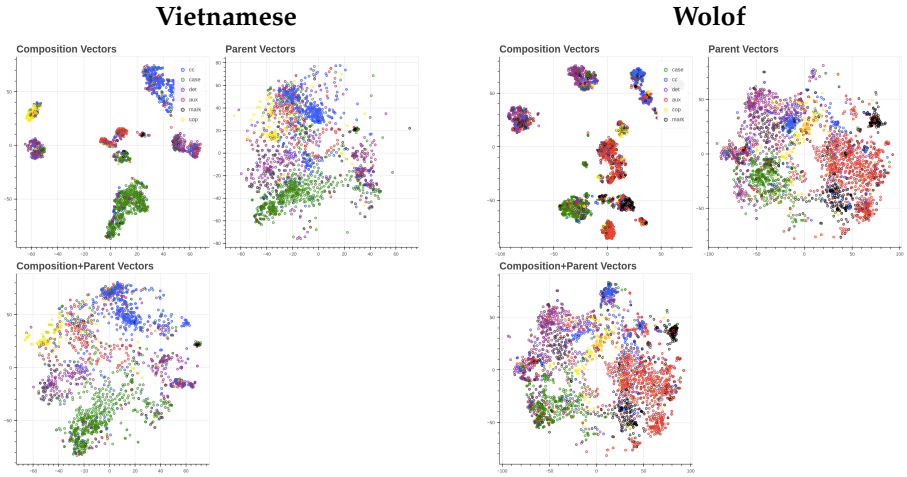


Swedish



Turkish





Acknowledgments

We are grateful to Miryam de Lhoneux, Artur Kulmizev, and Sara Stymne for valuable comments and suggestions. We thank the action editor and the three reviewers for constructive comments that helped us improve the final version. The research presented in this article was supported by the Swedish Research Council (grant 2016-01817).

References

Abney, Steven. 1991. Parsing by chunks. In Robert Berwick, Steven Abney, and Carol Tenny, editors, *Principle-Based Parsing*. Kluwer, pages 257–278.

Ballesteros, Miguel and Joakim Nivre. 2013. Getting to the roots of dependency parsing, 39:5–13.

Bārzdiņš, Guntis, Normunds Grūzītis, Gunta Nešpore, and Baiba Saulīte. 2007. Dependency-based hybrid model of syntactic analysis for the languages with a rather free word order. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 13–20.

Basirat, Ali and Joakim Nivre. 2021. Syntactic nuclei in dependency parsing—a multilingual exploration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1376–1387.

Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Belinkov, Yonatan. 2018. *On Internal Language Representations in Deep Learning: An Analysis of Machine Translation and Speech Recognition*. Ph.D. thesis, Massachusetts Institute of Technology.

Bharati, Akshar, Samar Husain, Dipti Misra, and Rajeev Sangal. 2009. Two stage constraint based hybrid approach to free word order language dependency parsing. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT)*, pages 77–80. `https://doi.org/10.3115/1697236.1697251`

Bharati, Akshar and Rajeev Sangal. 1993. Parsing free word order languages in the Paninian framework. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 105–111. `https://doi.org/10.3115/981574.981589`

Böhmová, Alena, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague Dependency Treebank: A three-level annotation scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127.

Buchholz, Sabine, Jorn Veenstra, and Walter Daelemans. 1999. Cascaded grammatical relation assignment. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*, pages 239–246.

Dyer, Chris, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational*

Linguistics (ACL), pages 334–343. https://doi.org/10.3115/v1/P15-1033

Falenska, Agnieszka and Jonas Kuhn. 2019. The (non-)utility of structural features in BiLSTM-based dependency parsers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 117–128. https://doi.org/10.18653/v1/P19-1012

Finlayson, Matthew, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843.

Han, Ji Yoon, Tae Hwan Oh, Lee Jin, and Hansaem Kim. 2020. Annotation issues in Universal Dependencies for Korean and Japanese. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 99–108.

Hewitt, John and Percy Liang. 2019. Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*.

Hupkes, Dieuwke, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926. https://doi.org/10.1613/jair.1.11196

Järvinen, Timo and Pasi Tapanainen. 1998. Towards an implementable dependency grammar. In *Proceedings of the Workshop on Processing of Dependency-Based Grammars (ACL-COLING)*, pages 1–10.

Johnson, Paul and Holger Schielzeth. 2017. The coefficient of determination r2 and intra-class correlation coefficient from generalized linear mixed-effects models revisited and expanded. *Journal of the Royal Society Interface*, 14:20170213.

Kahane, Sylvain. 1997. Bubble trees and syntactic representations. In *Proceedings of the 5th Meeting of Mathematics of Language*, pages 70–76.

Kiperwasser, Eliyahu and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327. https://doi.org/10.1162/tacl_a_00101

Kudo, Taku and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Workshop on Computational Language Learning (CoNLL)*, pages 63–69. https://doi.org/10.3115/1118853.1118869

Kuhlmann, Marco, Carlos Gómez-Rodríguez, and Giorgio Satta. 2011. Dynamic programming algorithms for transition-based dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 673–682.

de Lhoneux, Miryam, Miguel Ballesteros, and Joakim Nivre. 2019. Recursive subtree composition in LSTM-based dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1566–1576.

de Lhoneux, Miryam, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017. From raw text to Universal Dependencies—Look, no tags! In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 207–217.

de Lhoneux, Miryam, Sara Stymne, and Joakim Nivre. 2017. Arc-hybrid non-projective dependency parsing with a static-dynamic oracle. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 99–104.

de Lhoneux, Miryam, Sara Stymne, and Joakim Nivre. 2020. What should/do/can LSTMs learn when parsing auxiliary verb constructions? *Computational Linguistics*, 46(4):763–784.

Van der Maaten, Laurens and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11):2579–2605.

de Marneffe, Marie, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational Linguistics*, 47:255–308. https://doi.org/10.1162/coli_a_00402

Mohammadshahi, Alireza and James Henderson. 2020. Graph-to-graph transformer for transition-based dependency parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3278–3289.

Nespore, Gunta, Baiba Saulite, Guntis Barzdins, and Normunds Gruzitis. 2010. Comparison of the SemTi-Kamols and Tesnière's dependency grammars. In *Proceedings of the 4th International*

Conference on Human Language Technologies—the Baltic Perspective, pages 233–240.

Nivre, Joakim. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 149–160.

Nivre, Joakim. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together (ACL)*, pages 50–57.

Nivre, Joakim. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34:513–553. https://doi.org/10.1162/coli.07-056-R1-07-027

Nivre, Joakim. 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 (ACL-IJCNLP)*, pages 351–359. https://doi.org/10.3115/1687878.1687929

Nivre, Joakim and Chiao-Ting Fang. 2017. Universal dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 86–95.

Nivre, Joakim, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 2216–2219.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Dan Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, pages 1659–1666.

Nivre, Joakim, 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 evergrowing multilingual treebank collection. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC)*, pages 4034–4043.

Samuelsson, Christer. 2000. A statistical theory of dependency syntax. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, pages 684–690. https://doi.org/10.3115/992730.992745

Sangati, Federico and Chiara Mazza. 2009. An English dependency treebank à la Tesnière. In *Proceedings of the 8th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 173–184.

Sgall, Petr, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Pragmatic Aspects*. Reidel.

Shi, Tianze, Liang Huang, and Lillian Lee. 2017. Fast(er) exact decoding and global training for transition-based dependency parsing via a minimal feature set. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 12–23. https://doi.org/10.18653/v1/D17-1002

Smith, Aaron, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 treebanks, 34 models: Universal dependency parsing with multi-treebank models. In *Proceedings of the 2018 CoNLL Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123.

Stenetorp, Pontus. 2013. Transition-based dependency parsing using recursive neural networks. In *NIPS Workshop on Deep Learning*.

Tesnière, Lucien. 1959. *Éléments de syntaxe structurale*. Editions Klincksieck.

Yamada, Hiroyasu and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 195–206.

Zeman, Daniel, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, Hórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Deniz Baran Aslan, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad

Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Mihaela Cristescu, Philemon Daniel, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọ́ájídé Ishola, Kaoru Ito, Tomáš Jelínek, Apoorva Jha, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K. Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Boris Katz, Tolga Kayadelen, Jessica Kenney, Vàclava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen,

Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Dmitry Sichinava, Janine Siewert, Einar Freyr Sigurðsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Rachele Sprugnoli, Steinþór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2021. Universal dependencies 2.8.1. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Zhang, Yue and Joakim Nivre. 2011. Transition-based parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 188–193.