

# Syntactic Nuclei in Dependency Parsing – A Multilingual Exploration

**Ali Basirat**

Uppsala University  
Dept. of Linguistics and Philology  
ali.basirat@lingfil.uu.se

**Joakim Nivre**

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

## Abstract

Standard models for syntactic dependency parsing take words to be the elementary units that enter into dependency relations. In this paper, we investigate whether there are any benefits from enriching these models with the more abstract notion of nucleus proposed by Tesnière. We do this by showing how the concept of nucleus can be defined in the framework of Universal Dependencies and how we can use composition functions to make a transition-based dependency parser aware of this concept. Experiments on 12 languages show that nucleus composition gives small but significant improvements in parsing accuracy. Further analysis reveals that the improvement mainly concerns a small number of dependency relations, including relations of coordination, direct objects, nominal modifiers, and main predicates.

## 1 Introduction

A syntactic dependency tree consists of directed arcs, representing syntactic relations like subject and object, connecting a set of nodes, representing the elementary syntactic units of a sentence. In contemporary dependency parsing, it is generally assumed that the elementary units are word forms or tokens, produced by a tokenizer or word segmenter. A consequence of this assumption is that the shape and size of dependency trees will vary systematically across languages. In particular, morphologically rich languages will typically have fewer elementary units and fewer relations than more analytical languages, which use independent function words instead of morphological inflection to encode grammatical information. This is illustrated in Figure 1, which contrasts two equivalent sentences in English and Finnish, annotated with dependency trees following the guidelines of Universal Dependencies (UD) (Nivre et al., 2016, 2020), which assume words as elementary units.

An alternative view, found in the seminal work of Tesnière (1959), is that dependency relations hold between slightly more complex units called *nuclei*, semantically independent units consisting of a content word together with its grammatical markers, regardless of whether the latter are realized as independent words or not. Thus, 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 an auxiliary verb. The latter type is known as a *dissociated nucleus*. If we assume that the elementary syntactic units of a dependency tree are nuclei rather than word forms, then the English and Finnish sentences will have the same dependency trees, visualized in Figure 2, and will differ only in the realization of their nuclei. In particular, while nominal nuclei in Finnish are consistently realized as single nouns inflected for case, the nominal nuclei in English involve standalone articles and the preposition *from*.

In this paper, we set out to investigate whether research on dependency parsing can benefit from making explicit use of Tesnière’s notion of nucleus, from the point of view of accuracy, interpretability and evaluation. We do this from a multilingual perspective, because it is likely that the effects of introducing nuclei will be different in different languages, and we strongly believe that a comparison between different languages is necessary in order to assess the potential usefulness of this notion. We are certainly not the first to propose that Tesnière’s notion of nucleus can be useful in parsing. One of the earliest formalizations of dependency grammar for the purpose of statistical parsing, that of Samuelsson (2000), had this notion at its core, and Sangati and Mazza (2009) presented a conversion of the Penn Treebank of English to Tesnière style

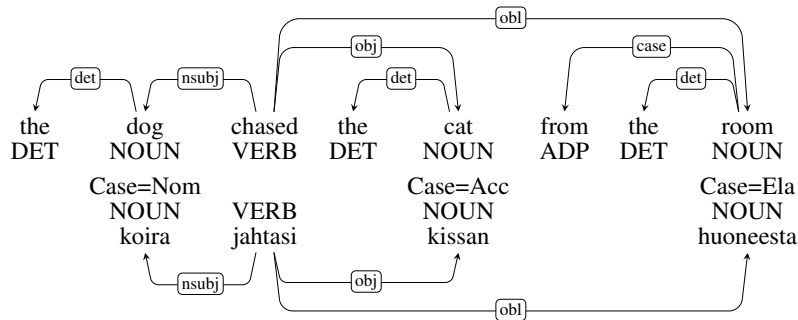


Figure 1: Word-based dependency trees for equivalent sentences from English (top) and Finnish (bottom).

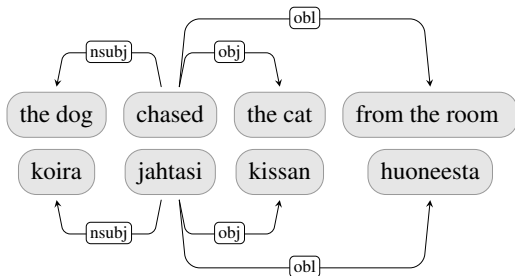


Figure 2: Nucleus-based dependency trees for equivalent sentences from English (top) and Finnish (bottom).

representations, including nuclei. However, previous attempts have been hampered by the lack of available parsers and resources to test this hypothesis on a large scale. Thus, the model of Samuelsson (2000) was never implemented, and the treebank conversion of Sangati and Mazza (2009) is available only for English and in a format that no existing dependency parser can handle. We propose to overcome these obstacles in two ways. On the resource side, we will rely on UD treebanks and exploit the fact that, although the annotation is word-based, the guidelines prioritize dependency relations between content words that are the cores of syntactic nuclei, which facilitates the recognition of dissociated nuclei and gives us access to annotated resources for a wide range of languages. On the parsing side, we will follow a transition-based approach, which can relatively easily be extended to include operations that create representations of syntactic nuclei, as previously shown by de Lhoneux et al. (2019a), something that is much harder to achieve in a graph-based approach.

## 2 Related Work

Dependency-based guidelines for syntactic annotation generally discard the nucleus as the basic syntactic unit in favor of the (orthographic) word form, possibly with a few exceptions for fixed

multiword expressions. A notable exception is the three-layered annotation scheme of the Prague Dependency Treebank (Hajič et al., 2000), where nucleus-like concepts are captured at the tectogrammatical level according to the Functional Generative Description (Sgall et al., 1986). Bärzdīņš et al. (2007) propose a syntactic analysis model for Latvian based on the x-word concept analogous to the nucleus concept. In this grammar, 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, as already mentioned, Sangati and Mazza (2009) develop an algorithm to convert English phrase structure trees to Tesnière style representations.

When it comes to syntactic parsing, Järvinen and Tapanainen (1998) were pioneers in adapting 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. As an alternative to the rule-based approach of Järvinen and Tapanainen (1998), Samuelsson (2000) defined a generative statistical model for nucleus-based dependency parsing, which however was never implemented.

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 et al. (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, Tongchim

et al. (2008) show that base-NP chunking can significantly improve the accuracy of dependency parsing for Thai, and Durgar El-Kahlout et al. (2014) show that chunking improves dependency parsing of Turkish. Das et al. (2016) study the importance of chunking in the transfer parsing model between Hindi and Bengali, and Lacroix (2018) show that NP chunks are informative for universal part-of-speech tagging and dependency parsing.

In a more recent study, de Lhoneux et al. (2019b) investigate whether the hidden representations of a neural transition-based dependency parser encodes information about syntactic nuclei, with special reference to verb groups. 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 et al. (2019a). 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 this paper, we will exploit variations of this technique to create parser-internal representations of syntactic nuclei, as discussed in Section 4. However, first we need to discuss how to identify nuclei in UD treebanks.

### 3 Syntactic Nuclei in UD

UD<sup>1</sup> (Nivre et al., 2016, 2020) is an ongoing project aiming to provide cross-linguistically consistent morphosyntactic annotation of many languages around the world. The latest release (v2.7) contains 183 treebanks, representing 104 languages and 20 language families. The syntactic annotation in UD is based on dependencies 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.<sup>2</sup> 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.<sup>3</sup>

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* with functional relations drawn below the sentence and other relations above. 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 discontinuous, 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<sup>4</sup> 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 (*flickan*), and not at all in Finnish.
- 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 relative 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 which 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 TAME marker and a verbal predicate. An example is the English verb group *will be causing* in Figure 3, which

<sup>1</sup><https://universaldependencies.org>

<sup>2</sup>Except in some cases of ellipsis, like *she did*, where the auxiliary verb *did* is “promoted” to form a nucleus on its own.

<sup>3</sup>Again, there are a few well-defined exceptions to the rule that function words are leaves, including ellipsis, coordination, and fixed multiword expressions.

<sup>4</sup>A more detailed description of the relations is available in the UD documentation at <https://universaldependencies.org>.

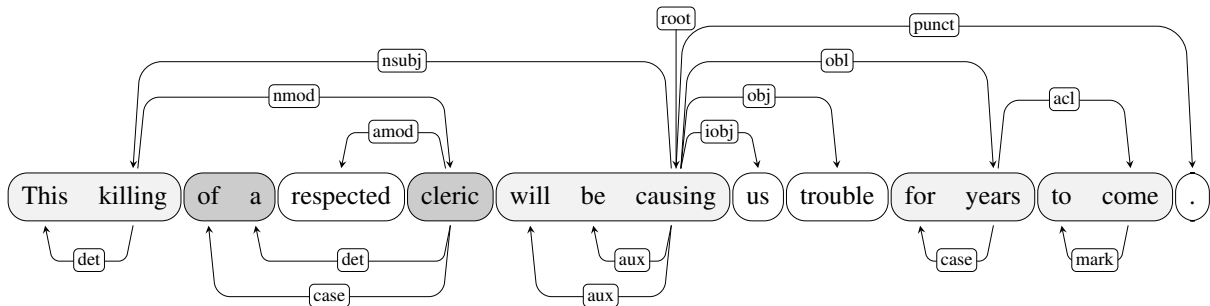


Figure 3: Syntactic UD representation with functional relations drawn below the sentence. Dissociated nuclei are grayed, with a darker shade for the discontinuous nucleus.

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 lučšij 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 probably 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, we 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.<sup>5</sup>

<sup>5</sup>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

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, 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 not (yet) available. Moreover, evaluation results would not be comparable to previous research. Therefore, we will make use of the nucleus concept in UD in three more indirect ways:

- **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.<sup>6</sup>
- **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 below.
- **Oracle Parsing:** To establish an upper bound on what a nucleus-aware parser can achieve, we will create a version of the UD representation which is still a word-based dependency

(*aux*, *cop* and *mark*).

<sup>6</sup>Our use of CLAS differs only in that we include punctuation in the evaluation, whereas Nivre and Fang (2017) excluded it.

tree, but where nuclei are explicitly represented by letting the word form for each nucleus core be a concatenation of all the word forms that are part of the nucleus.<sup>7</sup> We call this oracle parsing to emphasize that the parser has oracle information about the nuclei of a sentence, although it still has to predict all the syntactic relations.

#### 4 Syntactic Nuclei in Transition-Based Dependency Parsing

A transition-based dependency parser derives a dependency tree from the sequence of words forming a sentence (Yamada and Matsumoto, 2003; Nivre, 2003, 2004). The parser constructs the 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  $A$ .

For the experiments in this paper, we use a version of the arc-hybrid transition system initially proposed by Kuhlmann et al. (2011), where the initial configuration has all words  $w_1, \dots, w_n$  plus an artificial root node  $r$  in  $B$ , while  $S$  and  $A$  are empty.<sup>8</sup> 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.<sup>9</sup>

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  $x_1, \dots, x_n$  representing the input words  $w_1, \dots, w_n$  and feeds these vectors through a BiLSTM that outputs contextu-

alized word vectors  $v_1, \dots, v_n$ , which are stored in the buffer  $B$ . Parsing is then performed by iteratively applying the transition predicted by an 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 paper, 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 et al., 2017; Falenska and Kuhn, 2019).

The transition-based parser as described so far 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  $v_h$  representing the syntactic head is retained in  $S$  or  $B$ , while the vector  $v_d$  representing the syntactic dependent is removed from  $S$ . In order to make the parser sensitive to (dissociated) nuclei in its internal representations, we follow de Lhoneux et al. (2019a) 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 label  $l$ , we replace the current representation of  $h$  with the output of a function  $f(h, d, l)$ . We can then control the information flow for nuclei and other constructions through the definition of  $f(h, d, l)$ .

**Hard Composition:** The simplest version, which we call *hard* composition, is to explicitly condition the composition on the dependency label  $l$ . In this setup,  $f(h, d, l)$  combines the head and dependent vectors only if  $l$  is a functional relation and simply returns the head vector otherwise:

$$f(h, d, l) = \begin{cases} \vec{h} \circ \vec{d} & \text{if } l \in F \\ \vec{h} & \text{otherwise} \end{cases} \quad (1)$$

We use  $\vec{x}$  to denote the vector representation of  $x$ <sup>10</sup> and  $F$  to denote the set of seven functional relations defined in Section 3. The composition operator  $\circ$  can be any function of the form  $\mathbb{R}^n \times \mathbb{R}^n \rightarrow$

<sup>7</sup>The English sentence in Figure 1 thus becomes: *the dog-the chased the cat-the from the room-the-from*.

<sup>8</sup>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).

<sup>9</sup>This extension of the arc-hybrid system was proposed by de Lhoneux et al. (2017b), inspired by the corresponding extension of the arc-standard system by Nivre (2009).

<sup>10</sup>In the baseline parser,  $\vec{h}$  is always identical to the contextualized representation  $v_h$  of the head word  $w_h$ , but after introducing composition operations we need a more abstract notation.

$\mathbb{R}^n$ , e.g., vector addition  $\vec{h} + \vec{d}$ , where  $n$  is the dimensionality of the vector space.

**Soft Composition:** The *soft* composition is similar to the hard composition, but instead of applying the composition operator to the head and dependent vectors, the operator is applied to the head vector and a vector representation of the entire dependency arc  $(h, d, l)$ . The vector representation of the dependency arc is trained by a differentiable function  $g$  that encodes the dependency label  $l$  into a vector  $\vec{l}$  and maps the triple  $(\vec{h}, \vec{d}, \vec{l})$  to a vector space, i.e.,  $g : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  where  $n$  and  $m$  are the dimensionalities of the word and label spaces, respectively. An example of  $g$  is a perceptron with a sigmoid activation that maps the vector representations of  $h$ ,  $d$  and  $l$  to a vector space:

$$g(h, d, l) = \sigma(W(\vec{h} \odot \vec{d} \odot \vec{l}) + b) \quad (2)$$

where  $\odot$  is the vector concatenation operator. The soft nucleus composition is then:

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

The parameters of the function  $g$  are trained with the other parameters of the parser.

**Generalized Composition:** To test our hypothesis that composition is beneficial for dissociated nuclei, we contrast both hard and soft composition to a generalized version of soft composition, where we do not restrict the application to functional relations. In this case, the composition function is:

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

where  $l$  can be any dependency label. In this approach, the if-clause in Equation 2 and 3 is eliminated and the parser itself learns in what conditions the composition should be performed. In particular, if the composition operator is addition, and  $g$  is a perceptron with a sigmoid activation on the output layer (as in Equation 2), then  $g$  operates as a gate that controls the contribution of the dependency elements  $h$ ,  $d$ , and  $l$  to the composition. If the composition should not be performed, it returns a vector close to zero.

## 5 Experiments

In the previous sections, we have shown how syntactic nuclei can be identified in the UD annotation and how transition-based parsers can be made

sensitive to these structures in their internal representations through the use of nucleus composition. We now proceed to a set of experiments investigating the impact of nucleus composition on a diverse selection of languages.

### 5.1 Experimental Settings

We use UUParser (de Lhoneux et al., 2017a; Smith et al., 2018), an evolution of the transition-based dependency parser of Kiperwasser and Goldberg (2016), which was the highest ranked transition-based dependency parser in the CoNLL shared task on universal dependency parsing in 2018 (Zeman et al., 2018). As discussed in Section 4, this is a greedy transition-based parser based on the extended arc-hybrid system of de Lhoneux et al. (2017b). It uses an MLP with one hidden layer to predict transitions between parser configurations, based on vectors representing two items on the stack  $S$  and one item in the buffer  $B$ . In the baseline model, these items are contextualized word representations produced by a BiLSTM with two hidden layers. The input to the BiLSTM for each word is the concatenation of a randomly initialized word embedding and a character-based representation produced by running a BiLSTM over the character sequence of the word. We use a dimensionality of 100 for the word embedding as well as for the output of the character BiLSTM.

For parsers with composition, we considered various composition operators  $\circ$  and functions  $g$ . For the former, we tested vector addition, vector concatenation,<sup>11</sup> and perceptron. For the latter we tried a multi-layer perceptron with different activation functions. Based on the results of the preliminary experiments, we selected vector addition for the composition operator  $\circ$  and the perceptron with sigmoid activation for the soft composition function  $g$ . The inputs to the perceptron consist of two token vectors of size 512 and a relation vector of size 10. The token vectors are the outputs of the BiLSTM layer of the parser and the relation vector is trained by a distinct embedding layer.

All parsers are trained for 50 epochs and all reported results are averaged over 10 runs with different random seeds. Altogether we explore five different parsers:

<sup>11</sup>The concatenation operator requires special care to keep vector dimensionality constant. We double the dimensionality of the contextual vectors and fill the extra dimensions with zeros. We then replace the zero part of the second operand with the first operand’s non-zero part at composition time.

Language	Treebank	Family	Size	LAS					CLAS				
				Base	Hard	Soft	Gen	Ora	Base	Hard	Soft	Gen	Ora
Arabic	PADT	AA-Semitic	242K	78.0	78.0	78.4	78.5	80.5	74.4	74.5	75.0	75.1	75.8
Basque	BDT	Basque	70K	73.5	73.6	74.1	75.0	78.9	70.9	71.1	71.5	72.6	75.3
Chinese	GSD	Sino-Tibetan	49K	72.1	72.1	72.8	72.8	80.4	68.8	68.6	69.5	69.5	74.6
English	EWT	IE-Germanic	188K	82.5	82.4	82.7	82.6	85.6	78.8	78.7	79.0	78.9	81.0
Finnish	TDT	Uralic	168K	78.5	78.7	79.5	80.1	84.2	77.5	77.7	78.6	79.1	81.8
Hebrew	HTB	AA-Semitic	108K	81.5	81.5	81.9	82.3	83.3	74.5	74.5	75.1	75.6	75.9
Hindi	HDTB	IE-Indic	294K	87.9	88.2	88.4	89.4	89.8	83.9	84.1	84.5	85.8	85.0
Italian	ISDT	IE-Romance	252K	87.4	87.6	87.8	87.8	89.0	81.5	81.6	82.1	82.1	82.8
Japanese	GSD	Japanese	61K	93.4	93.4	93.6	93.5	94.1	89.7	89.8	90.1	89.9	90.5
Korean	GSD	Korean	21K	75.1	75.0	75.4	75.8	76.6	74.9	74.8	75.2	75.7	75.7
Swedish	Talbanken	IE-Germanic	61K	76.9	77.2	77.5	78.0	82.9	73.2	73.4	73.9	74.5	78.2
Turkish	IMST	Turkic	40K	55.6	55.7	56.3	56.6	58.6	53.4	53.6	54.3	54.6	54.8
Average				78.5	78.6	79.0	79.4	82.0	75.1	75.2	75.7	76.1	77.6

Table 1: Parsing accuracy for 5 parsing models evaluated on 12 UD treebanks. Language family includes genus according to WALS for large families (AA = Afro-Asian, IE = Indo-European). LAS = Labeled Attachment Score. CLAS = Content Labeled Attachment Score.

- **Base(line)**: No composition.
- **Hard**: Baseline + hard composition.
- **Soft**: Baseline + soft composition.
- **Gen(eralized)**: Baseline + gen. composition.
- **Ora(ble)**: Baseline trained and tested on explicit annotation of nuclei (see Section 3).

Our experiments are carried out on a typologically diverse set of languages with different degrees of morphosyntactic complexity, as shown in Table 1. The corpus size is the total number of words in each treebank. We use UD v2.3 with standard data splits (Nivre et al., 2018). All evaluation results are on the development sets.<sup>12</sup>

## 5.2 Results

Table 1 reports the parsing accuracy achieved with our 5 parsers on the 12 different languages, using the standard LAS metric as well as the nucleus-aware CLAS metric. First of all, we see that hard composition is not very effective and mostly gives results in line with the baseline parser, with significant improvements only for Finnish and Hindi (two-tailed  $t$ -test,  $p = .05$ ). By contrast, soft composition improves accuracy for all languages and the improvements are statistically significant for both LAS and CLAS except for LAS in English ( $p = 0.06$ ). The average improvement is 0.5 and 0.6 percentage points for LAS and CLAS,

<sup>12</sup>Since we want to perform an informative error analysis, we avoid using the dedicated test sets.

respectively, which indicates that most of the improvement occurs on nucleus-external relations thanks to a more effective internal representation of dissociated nuclei. Generalized composition, finally, where we allow composition also for non-functional relations, is again significantly better than the baseline for all languages with respect to CLAS and for all languages except English with respect to LAS ( $p = 0.18$ ). For six languages (Basque, Finnish, Hebrew, Hindi, Korean and Swedish), generalized composition also gives significant improvements over soft composition (for both LAS and CLAS), but for the remaining languages there is little or no improvement.<sup>13</sup> A tentative conclusion is therefore that, while composition is almost universally beneficial when applied to nucleus-internal relations, its usefulness for other types of syntactic constructions is more sensitive to language-specific properties and needs further study.

Before we try to analyze the results in more detail, it is worth noting that most of the improvements due to composition are considerably below the improvements of the oracle parser.<sup>14</sup> However, it is important to keep in mind that, whereas the behavior of a composition parser is only affected

<sup>13</sup>For Japanese, there is even a significant decrease in CLAS.

<sup>14</sup>The exceptions are found in the CLAS results for generalized composition, which are better than the oracle parser for Hindi, equal for Korean, and fairly close for Japanese and Turkish.

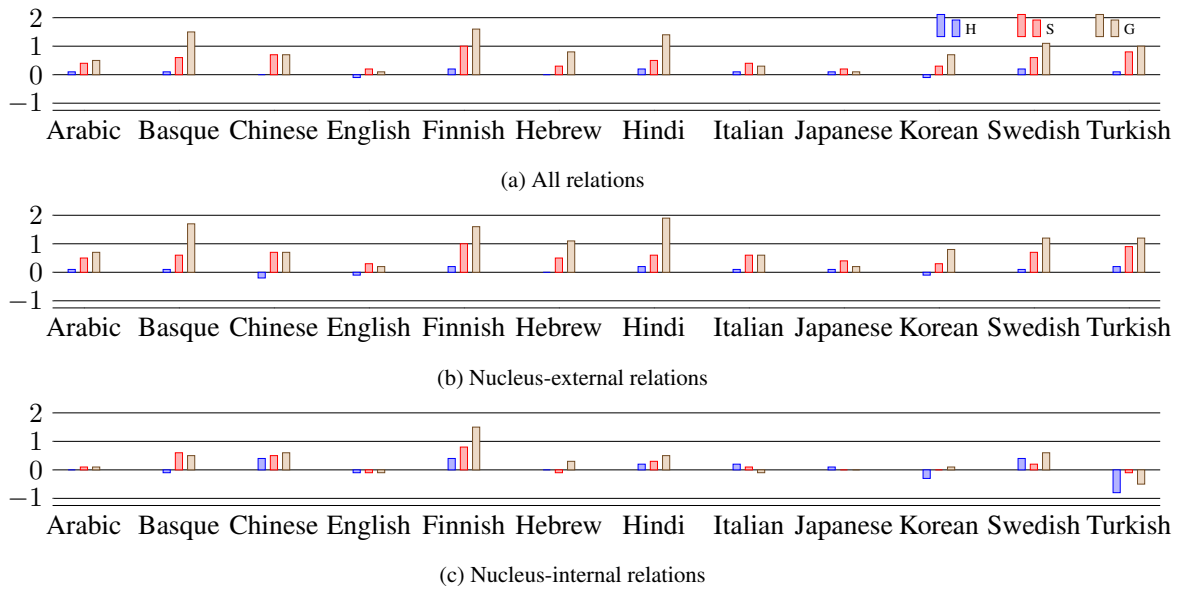


Figure 4: Absolute improvement (or degradation) in labeled F-score with respect to the baseline for hard (H), soft (S) and generalized (G) composition for different sets of relations: (a) all relations (corresponding to LAS scores), (b) nucleus-external relations (corresponding to CLAS scores), (c) nucleus-internal relations.

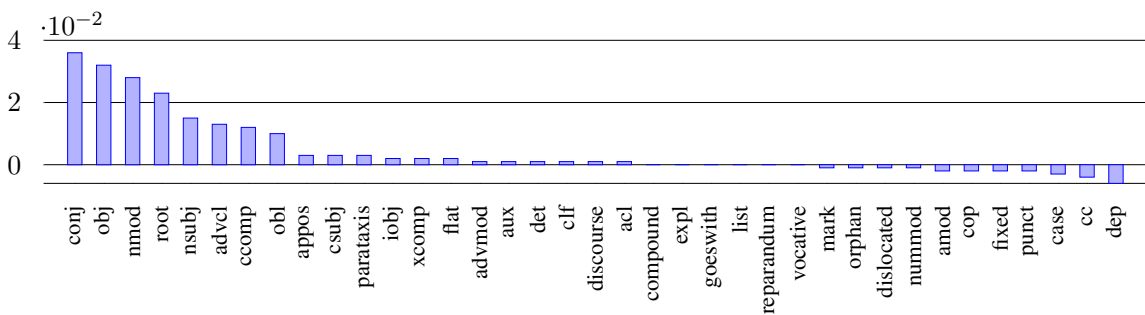


Figure 5: Improvement (or degradation) in labeled F-score with respect to the baseline for different UD relations (soft compositions), weighted by the relative frequency of each relation and averaged across all languages.

after a nucleus has been constructed, the oracle parser improves also with respect to the prediction of the nuclei themselves. This explains why the oracle parser generally improves more with respect to LAS than CLAS, by about one percentage point on average but by as much as 2.5 points for Chinese and 1.6 points for Turkish.

Figure 4 visualizes the impact of hard, soft and generalized nucleus composition for different languages, with a breakdown into (a) all relations, which corresponds to the difference in LAS compared to the baseline, (b) nucleus-external relations, which corresponds to the difference in CLAS, and (c) nucleus-internal relations. Overall, these graphs are consistent with the hypothesis that using composition to create parser-internal representations of (dissociated) nuclei primarily affects the prediction of nucleus-external relations, as the (a) and (b)

graphs are very similar and the (c) graphs mostly show very small differences. There are, however, a few notable exceptions. For Finnish, all three composition methods clearly improve the prediction of nucleus-internal relations as well as nucleus-external relations, by over 1 F-score point for generalized composition. Similar patterns, with smaller magnitude, can be seen for Chinese and Hindi, possibly also for Basque and Swedish. Conversely, for Turkish, especially hard and generalized composition has a detrimental effect on the prediction of nucleus-internal relations. Further research will be needed to explain what causes these deviant patterns.

To better understand the impact of nucleus composition, we now look at the effect on individual dependency relations. We focus on the soft composition model, which most directly implements the



Finnish		Chinese		Swedish		Turkish		Basque	
conj	0.09	nsubj	0.08	obl	0.06	conj	0.12	conj	0.06
obl	0.06	case	0.04	amod	0.06	root	0.08	advcl	0.05
nmod	0.06	nmod	0.04	advmod	0.06	obj	0.07	nmod	0.03
obj	0.04	ccomp	0.04	xcomp	0.05	ccomp	0.02	obj	0.02
root	0.03	obl	0.04	obj	0.04	fixed	0.01	xcomp	0.02
nsubj	0.02	obj	0.03	root	0.04	discourse	0.01	root	0.02
advcl	0.02	acl	0.02	conj	0.03	cc	0.01	ccomp	0.02
amod	0.02	advmod	0.02	nmod	0.03	amod	0.01	iobj	0.01
ccomp	0.02	det	0.02	acl	0.02	det	0.01	advmod	0.01
acl	0.01	root	0.02	advcl	0.01	advmod	0.01	flat	0.01

Table 2: Improvement (or degradation) in labeled F-score, weighted by relative frequency, for the 10 best UD relations in the 5 languages with greatest LAS improvements over the baseline (soft composition).

idea of nucleus composition (as opposed to composition in general) and which also provides the most stable improvements across languages. Figure 5 shows the improvement (or degradation) for individual UD relations, weighted by relative frequency and averaged over all languages. The most important improvements are observed for *conj*, *obj*, *nmod*, and *root*. The *conj* relation holds between conjuncts in a coordination structure; *obj* is the direct object relation; the *nmod* relation covers all nominal modifiers inside noun phrases, including prepositional phrase modifiers; and the *root* relation is assigned to the main predicate of a sentence. In addition, we see substantial improvements (of at least 1 F-score point) for four major clause relations – *nsubj* (nominal subjects), *advcl* (adverbial clauses), *ccomp* (complement clauses), and *obl* (oblique modifiers) – as well as smaller improvements for a long tail of additional relations. Of these, only *aux*, *clf*, and *det* are nucleus-internal relations and these three relations show very marginal improvements, so the results further support the hypothesis that richer internal representations of (dissociated) nuclei primarily improve the prediction of nucleus-external dependency relations, especially major clause relations.

It is important to remember that the results in Figure 5 are averaged over all languages and may hide interesting differences between languages. A full investigation of this variation is beyond the scope of this paper, but Table 2 presents a further zoom-in by presenting statistics on the top 10 relations in the 5 languages where LAS improves the most compared to the baseline. To a large extent, we find the same relations as in the aggregated statistics, although with slightly different orders, but there

are also interesting language-specific patterns. For Chinese one of the relations tied for second place is the nucleus-internal *case* relation. For Swedish one of the relations tied for second place is *amod* (adjectival modifier), which does not show positive improvements on average (although it is in the top ten also for Finnish and Turkish). These differences definitely deserve further investigation.

## 6 Conclusion

We have explored how the 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 conclude that the use of composition operations for building internal representations of syntactic nuclei, in particular the technique that we have called soft composition, can lead to small but significant improvements in parsing accuracy for nucleus-external relations, notably for relations of coordination, direct objects, nominal modifiers and main predicates. For some languages, further improvements can be obtained by applying composition also to the nucleus-external relations themselves (generalized composition). In future work we want to study the behavior of different types of nuclei in more detail, in particular how the different internal relations of nominal and verbal nuclei contribute to overall parsing accuracy. We also want to analyze the variation between different languages in more detail and see if it can be explained in terms of typological properties.

## Acknowledgments

We thank Daniel Dakota, Artur Kulmizev, Sara Stymne and Gongbo Tang for useful discussions and the EACL reviewers for constructive criticism. The research presented in this paper was supported by grant 2016-01817 of the Swedish Research Council. We acknowledge the computational resources provided by CSC in Helsinki and Sigma2 in Oslo through NeIC-NLPL ([www.nlpl.eu](http://www.nlpl.eu)).

## References

- S. Abney. Parsing by chunks. In R. Berwick, S. Abney, and C. Tenny, editors, *Principle-Based Parsing*, pages 257–278. Kluwer, 1991.
- M. Ballesteros and J. Nivre. Going to the roots of dependency parsing. *Computational Linguistics*, 39(1):5–13, 2013.
- G. Bärzdiņš, N. Grūzītis, G. Nešpore, and B. Saulīte. 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, Tartu, Estonia, May 2007. University of Tartu, Estonia.
- A. Bharati and R. Sangal. Parsing free word order languages in the paninian framework. In *Proceedings of the Annual Meeting of Association for Computational Linguistics*, pages 105—111, 1993.
- A. Bharati, S. Husain, D. Misra, and R. Sangal. Two stage constraint based hybrid approach to free word order language dependency parsing. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 77–80, 2009.
- S. Buchholz, J. Veenstra, and W. Daelemans. Cascaded grammatical relation assignment. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*, pages 239–246, 1999.
- A. Das, A. Saha, and S. Sarkar. Cross-lingual transfer parser from Hindi to Bengali using delexicalization and chunking. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 99–108, Varanasi, India, Dec. 2016. NLP Association of India.
- M. de Lhoneux, Y. Shao, A. Basirat, E. Kiperwasser, S. Stymne, Y. Goldberg, and J. Nivre. 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, 2017a.
- M. de Lhoneux, S. Stymne, and J. Nivre. Arc-hybrid non-projective dependency parsing with a static-dynamic oracle. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 99–104, Pisa, Italy, Sept. 2017b. Association for Computational Linguistics.
- M. de Lhoneux, M. Ballesteros, and J. Nivre. 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, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics.
- M. de Lhoneux, S. Stymne, and J. Nivre. What should/do/can LSTMs learn when parsing auxiliary verb constructions? *CoRR*, abs/1907.07950, 2019b. URL <http://arxiv.org/abs/1907.07950>.
- İ. Durgar El-Kahlout, A. A. Akın, and E. Yılmaz. Initial explorations in two-phase Turkish dependency parsing by incorporating constituents. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 82–89, Dublin, Ireland, Aug. 2014. Dublin City University.
- C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July 2015. Association for Computational Linguistics.
- A. Falenska and J. Kuhn. 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, 2019.
- J. Hajič, A. Böhmová, E. Hajičová, and B. Vidová-Hladká. The Prague Dependency Treebank: A Three-Level Annotation Scenario. In A. Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127. Amsterdam:Kluwer, 2000.
- T. Järvinen and P. Tapanainen. Towards an implementable dependency grammar. In *Processing of Dependency-Based Grammars*, 1998.
- E. Kiperwasser and Y. Goldberg. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327, 2016.
- T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, page 1–7, USA, 2002. Association for Computational Linguistics.

- M. Kuhlmann, C. Gómez-Rodríguez, and G. Satta. Dynamic programming algorithms for transition-based dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 673–682, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
- O. Lacroix. Investigating NP-chunking with universal dependencies for English. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 85–90, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics.
- G. Nespore, B. Saulite, G. Barzdins, and N. Gruzitis. Comparison of the semi-kamols and tesnière’s dependency grammars. In *Proceedings of the 4th International Conference on Human Language Technologies — the Baltic Perspective*, 2010.
- J. Nivre. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160, Nancy, France, Apr. 2003.
- J. Nivre. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- J. Nivre. 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, 2009.
- J. Nivre and C.-T. Fang. Universal dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 86–95, 2017.
- J. Nivre, M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajič, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty, and D. Zeman. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, 2016.
- J. Nivre, M. Abrams, Ž. Agić, L. Ahrenberg, L. Antonsen, K. Aplonova, M. J. Aranzabe, G. Arutie, M. Asahara, L. Ateyah, M. Attia, A. Atutxa, L. Augustinus, E. Badmaeva, M. Ballesteros, E. Banerjee, S. Bank, V. Barbu Mititelu, V. Basmov, J. Bauer, S. Bellato, K. Bengoetxea, Y. Berzak, I. A. Bhat, R. A. Bhat, E. Biagetti, E. Bick, R. Blokland, V. Bobicev, C. Börstell, C. Bosco, G. Bouma, S. Bowman, A. Boyd, A. Burchardt, M. Candito, B. Caron, G. Caron, G. Cebiroğlu Eryiğit, F. M. Cecchini, G. G. A. Celano, S. Čéplö, S. Cetin, F. Chalub, J. Choi, Y. Cho, J. Chun, S. Cinková, A. Collomb, Ç. Çöltekin, M. Connor, M. Courtin, E. Davidson, M.-C. de Marneffe, V. de Paiva, A. Diaz de Ilarraza, C. Dickerson, P. Dirix, K. Dobrovoljc, T. Dozat, K. Droganova, P. Dwivedi, M. Eli, A. Elkahky, B. Ephrem, T. Erjavec, A. Etienne, R. Farkas, H. Fernandez Alcalde, J. Foster, C. Freitas, K. Gajdošová, D. Galbraith, M. Garcia, M. Gärdenfors, S. Garza, K. Gerdes, F. Ginter, I. Goenaga, K. Gojenola, M. Gökırmak, Y. Goldberg, X. Gómez Guinovart, B. Gonzáles Saavedra, M. Grioni, N. Grüzitis, B. Guillaume, C. Guillot-Barbance, N. Habash, J. Hajič, J. Hajič jr., L. Hà Mý, N.-R. Han, K. Harris, D. Haug, B. Hladká, J. Hlaváčová, F. Hociung, P. Hohle, J. Hwang, R. Ion, E. Irimia, O. Ishola, T. Jelínek, A. Johannsen, F. Jørgensen, H. Kaşıkara, S. Kahane, H. Kanayama, J. Kanerva, B. Katz, T. Kayadelen, J. Kenney, V. Kettnerová, J. Kirchner, K. Kopacewicz, N. Kotsyba, S. Krek, S. Kwak, V. Laippala, L. Lambertino, L. Lam, T. Lando, S. D. Larasati, A. Lavrentiev, J. Lee, P. Lê Hông, A. Lenci, S. Lertpradit, H. Leung, C. Y. Li, J. Li, K. Li, K. Lim, N. Ljubešić, O. Loginova, O. Lya-shenskaya, T. Lynn, V. Macketanz, A. Makazhanov, M. Mandl, C. Manning, R. Manurung, C. Măranduc, D. Mareček, K. Marheinecke, H. Martínez Alonso, A. Martins, J. Mašek, Y. Matsumoto, R. McDonald, G. Mendonça, N. Miekka, M. Misirpashayeva, A. Missilä, C. Mititelu, Y. Miyao, S. Montemagni, A. More, L. Moreno Romero, K. S. Mori, S. Mori, B. Mortensen, B. Moskalevskiy, K. Muischnek, Y. Murawaki, K. Müürisep, P. Nainwani, J. I. Navarro Horňáček, A. Nedoluzhko, G. Nešpore-Běrzkalne, L. Nguyễn Thị, H. Nguyễn Thị Minh, V. Nikolaev, R. Nitisaroj, H. Nurmi, S. Ojala, A. Olúòkun, M. Omura, P. Osenova, R. Östling, L. Øvrelid, N. Partanen, E. Pascual, M. Passarotti, A. Patejuk, G. Paulino-Passos, S. Peng, C.-A. Perez, G. Perrier, S. Petrov, J. Piitulainen, E. Pitler, B. Plank, T. Poibeau, M. Popel, L. Pretkalniņa, S. Prévost, P. Prokopidis, A. Przepiórkowski, T. Puolakainen, S. Pyysalo, A. Rääbis, A. Rademaker, L. Ramasamy, T. Rama, C. Ramisch, V. Ravishankar, L. Real, S. Reddy, G. Rehm, M. Rießler, L. Rinaldi, L. Rituma, L. Rocha, M. Romanenko, R. Rosa, D. Rovati, V. Roşca, O. Rudina, J. Rueter, S. Sadde, B. Sagot, S. Saleh, T. Samardžić, S. Samson, M. Sanguinetti, B. Saulite, Y. Sawanakunanon, N. Schneider, S. Schuster, D. Seddah, W. Seeker, M. Seraji, M. Shen, A. Shimada, M. Shohibussirri, D. Sichinava, N. Silveira, M. Simi, R. Simionescu, K. Simkó, M. Šimková, K. Simov, A. Smith, I. Soares-Bastos, C. Spadine, A. Stella, M. Straka, J. Strnadová, A. Suhr, U. Sulubacak, Z. Szántó, D. Taji, Y. Takahashi, T. Tanaka, I. Tellier, T. Trosterud, A. Trukhina, R. Tsarfaty, F. Tyers, S. Uematsu, Z. Urešová, L. Uria, H. Uszkoreit, S. Vajjala, D. van Niekerk, G. van Noord, V. Varga, E. Villemonde de la Clergerie, V. Vincze, L. Wallin, J. X. Wang, J. N. Washington, S. Williams, M. Mirán, T. Woldemariam, T.-s. Wong, C. Yan, M. W. M. Yavrumyan, Z. Yu, Z. Žabokrtský, A. Zeldes, D. Zeman, M. Zhang, and H. Zhu. Universal dependencies 2.3, 2018. URL <http://hdl.handle.net/11234/1-2895>. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguis-

tics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

- J. Nivre, M.-C. de Marneffe, F. Ginter, J. Hajič, C. D. Manning, S. Pyysalo, S. Schuster, F. Tyers, and D. Zeman. Universal Dependencies v2: An ever-growing multilingual treebank collection. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC)*, 2020.
- C. Samuelsson. A statistical theory of dependency syntax. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*, 2000.
- F. Sangati and C. Mazza. An English dependency treebank à la tesnière. In *Proceedings of the 8th International Workshop on Treebanks and Linguistic Theories*, pages 173–184, Nov. 2009.
- P. Sgall, E. Hajičová, and J. Panevová. *The Meaning of the Sentence in Its Pragmatic Aspects*. Reidel, 1986.
- T. Shi, L. Huang, and L. Lee. 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, 2017.
- A. Smith, B. Bohnet, M. de Lhoneux, J. Nivre, Y. Shao, and S. Stymne. 82 treebanks, 34 models: Universal dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics.
- P. Stenetorp. Transition-based dependency parsing using recursive neural networks. In *NIPS Workshop on Deep Learning*, 2013.
- L. Tesnière. *Éléments de syntaxe structurale*. Editions Klincksieck, 1959.
- S. Tongchim, V. Sornlertlamvanich, and H. Isahara. Experiments in base-NP chunking and its role in dependency parsing for Thai. In *Coling 2008: Companion volume: Posters*, pages 123–126, Manchester, UK, Aug. 2008. Coling 2008 Organizing Committee.
- H. Yamada and Y. Matsumoto. Statistical dependency analysis with support vector machines. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 195–206, Nancy, France, Apr. 2003.
- D. Zeman, J. Hajič, M. Popel, M. Potthysast, M. Straka, F. Ginter, J. Nivre, and S. Petrov. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 2018.