Nucleus Composition in Transition-Based Dependency Parsing

Joakim Nivre[*†] **Ali Basirat**[‡] **Luise Dürlich**[*†] **Adam Moss**[*]
[*]Uppsala University, Department of Linguistics and Philology
[†]RISE Research Institutes of Sweden
[‡]Linkøping University, Department of Computer and Information Science

Relevant UniDive working groups: WG1, WG3

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 (left), 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 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 1 (right), and will differ only in the realization of the nuclei involved. Thus, while 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 articles and the preposition *from*.

In this article, we first show how 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. We then go on to describe how transition-based parsers, as previously shown by de Lhoneux et al. (2020), can relatively easily be extended to include operations that create internal representations of syntactic nuclei. We then perform an experimental evaluation of such a parser on a diverse sample of 20 languages and analyze (a) to what extent nucleus composition can improve parsing accuracy for different languages, (b) which linguistic constructions benefit from these improvements, (c) how we can explain the different rates of improvement across languages, and (d) what information is encoded in the nucleus representations created through composition.

2 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 morphosyntactic annotation for as many languages as possible. The latest release from November 2022 (v2.11) features 243 annotated corpora, representing 138 languages from 22 language families. The syntactic annotation in UD is based on dependency relations and the elementary syntactic units are assumed to be words, but the

[1]https://universaldependencies.org

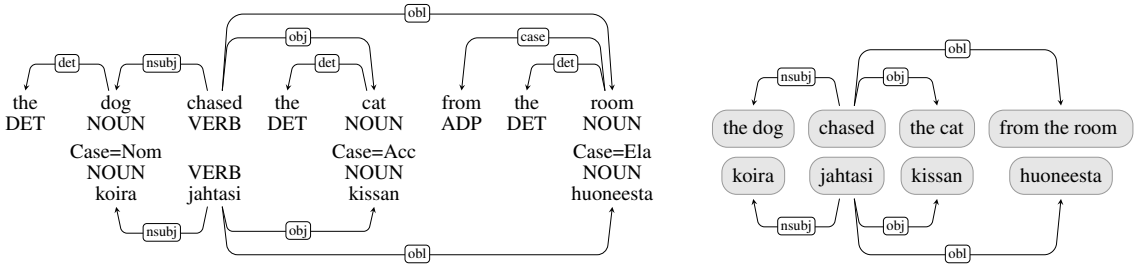

Figure 1: Word-based (left) and nucleus-based (right) dependency trees for equivalent sentences from English (top) and Finnish (bottom).

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 (almost) always go to the lexical core of a nucleus. 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. Given this type of representation, we can define a *nucleus* as a subtree where all internal dependencies are one of the following seven functional relations: **aux**, **case**, **cc**, **clf**, **cop**, **det** or **mark**.

3 Syntactic Nuclei in Transition-Based Dependency Parsing

As previously shown by 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.

We use a version of the influential parsing architecture of Kiperwasser and Goldberg (2016), with the arc-hybrid transition system initially proposed by Kuhlmann et al. (2011), extended with a so-called swap transition to allow the construction of non-projective dependencies de Lhoneux et al. (2017). In this parser, each word w_i on the stack S or buffer B is represented by a contextualized word vector $v_1, \ldots, v_n$, produced by a BiLSTM encoder applied to the sequence of input word embeddings $x_1, \ldots, x_n$. In the baseline parser, when two substructures headed by the words x_h and x_d are connected by dependency relation l in an 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 augment arc transitions with a composition operation, following de Lhoneux et al. (2019) and Basirat and Nivre (2021). 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 v_h the output of a learned function $g(v_h, v_d, v_l)$, which is the output of a (single-layered) perceptron with sigmoid activation applied to the concatenation of v_h, v_d and v_l.

4 Experiments and Analysis

To assess the impact of nucleus composition across languages with different typological properties, we perform an experimental study using data from 20 languages: Arabic, Armenian, Basque, Chinese, Finnish, Greek, Hebrew, Hindi, Indonesian, Irish, Italian, Japanese, Korean, Latvian, Persian, Russian, Swedish, Turkish, Vietnamese, and Wolof. 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, such as determiners. Analysis using dimensionality reduction and diagnostic classifiers suggests that nucleus composition increases the similarity of vectors representing nuclei of the same syntactic type.

Acknowledgments

This is an extended abstract of Nivre et al. (2022). We are grateful to Miryam de Lhoneux, Artur Kulmizev and Sara Stymne for valuable comments and suggestions. The research presented in the article was supported by the Swedish Research Council (grant 2016-01817).

References

Ali Basirat 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.

Miryam de Lhoneux, 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.

Miryam de Lhoneux, 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.

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

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

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Marco Kuhlmann, 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.

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

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

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34:513–553.

Joakim Nivre, Ali Basirat, Luise Dürlich, and Adam Moss. 2022. Nucleus composition in transition-based dependency parsing. *Computational Linguistics*, 48(4):849–886.

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

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Dan Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC)*, pages 4034–4043.

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

Hiroyasu Yamada 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.