Improving Surface-syntactic Universal Dependencies (SUD): surface-syntactic relations and deep syntactic features

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, Guy Perrier

Improving Surface-syntactic Universal Dependencies (SUD): surface-syntactic relations and deep syntactic features

Kim Gerdes
Almanach (Inria), LPP (CNRS)
Sorbonne Nouvelle
kim@gerdes.fr

Bruno Guillaume
Université de Lorraine, CNRS,
Inria, LORIA, Nancy, France
bruno.guillaume@inria.fr

Sylvain Kahane
Modyco,
Université Paris Nanterre & CNRS
sylvain@kahane.fr

Guy Perrier
Université de Lorraine, CNRS,
Inria, LORIA, Nancy, France
guy.perrier@loria.fr

Abstract

SUD is an annotation scheme for syntactic dependency treebanks, near isomorphic to UD (Universal Dependencies). Contrary to UD, it is based on syntactic criteria (favoring functional heads) and the relations are defined on distributional and functional bases. In this paper, we will recall and specify the general principles underlying SUD, present the updated set of SUD relations, discuss the central question of MWEs, and introduce an orthogonal layer of deep-syntactic features converted from the deep-syntactic part of the UD scheme.

1 Introduction

SUD (Surface-syntactic Universal Dependencies) is an annotation scheme that we proposed in a previous paper (Gerdes et al., 2018) as an alternative of the UD (Universal Dependencies) annotation scheme (Nivre and al., 2019). SUD follows surface syntax criteria (especially distributional criteria) and can be automatically converted into the UD scheme. SUD has now been used in the development of a treebank for Naija (Courtin et al., 2018; Caron et al., 2019) and treebanks for French and Chinese are in development. Some principles underlying SUD have been further clarified and will be exposed here.

Section 2 recalls and specifies the general principles of SUD. For a more detailed explanation of these principles, we refer the reader to the initial SUD presentation (Gerdes et al., 2018). The following sections present the original points of the article. Section 3 presents the set of SUD relations, which has been updated, providing a better distinction between surface syntactic and deep syntactic features (following the separation between surface and deep syntax of the Meaning-Text Theory (Mel'čuk, 1988)). Section 4 discusses the need for a separate encoding for MWEs' POS in SUD. Section 5 presents some principles of the UD $\Leftrightarrow$ SUD conversion.

2 General principles of SUD

2.1 Surface-syntactic criteria for heads

We will briefly recall the criteria for surface-syntactic headedness. These criteria have been the subject of much discussion (Hudson, 1984; Hudson, 1987; Mel'čuk, 1988). In the original paper (Gerdes et al., 2018), we retain two central criteria: First, the surface syntactic head of a unit **U** is an element of **U** that determines the distribution of **U**, that is, the syntactic position that **U** can occupy; for instance, *Mary* cannot be the head of **U** = *to Mary*, because *Mary* and **U** occupy completely different syntactic positions.[1] Such a criterion favors functional heads, while UD treats functional elements as leaves and

[1] One exception to this is the case wh-words: although *it is perfect* and *which is perfect* have different distributions (the relative clause modifies a noun), we decided not to take the wh-word at the syntactic head, but to favor its pronominal role inside the relative clause. This is not a theoretical choice, but rather a pragmatic decision preserving the tree structure.

poses as a principle that syntactic relations must be between content words, functional words being then relegated to being markers of the content words.

In some cases, the first criterion does not give a clear situation because two words have head features. In this case, a second gradual criterion comes into play where we prefer to give the status of dependent to the one that changes less the distribution of the unit. According to this principle, a coordinative conjunction such as *and* does not govern the conjunct following it, because *and Mary*, *and red*, or *and is sleeping* occupy completely different positions. In the same way, the determiner is analyzed as a dependent of the noun because nouns partly control the distribution of a combination determiner-noun (*this morning* can work as a modifier of a verb contrary to *this boy*).

A last point concerns coordination: SUD adopts a string-analysis of coordination, where each conjunct depends on the previous one, contrary to UD, which adopts a bouquet-analysis, where each conjunct depends on the first conjunct. One of the key arguments for the string-analysis is that it reduces the dependency length (Gibson, 1998; Liu, 2008; Futrell et al., 2015).

2.2 Criteria for SUD relations

SUD relations (that is, dependency labels) are defined by means of functional criteria: Two units that commute in the same syntactic position (and consequently bear the same function) must be linked to their governor by the same relation. The characterization of a relation is based on the whole paradigm of elements that can commute in the dependent position, while UD relations strongly rely on the POS of the dependent. For instance, a unique `comp:obj` relation for direct object complements is considered in SUD, where UD considers three relations: `obj` for a nominal object (*I imagine a dance*), `ccomp` for a clausal object (*I imagine (that) he dances*) and `xcomp` for a clausal object without its own subject (*I imagine to dance*).

This last relation raises another problem. (Przepiórkowski and Patejuk, 2018) extensively argue that UD's `xcomp` is particularly unsatisfactory because it is based on a property (not having its own subject), which is orthogonal to the syntactic function and can even be realized with modifiers (*He came without running*).[2] We make a clear distinction between surface-syntactic properties, which determine relation classes, and deep-syntactic properties, such as those expressed by `xcomp`. In Section 3.2, we will propose to represent deep-syntactic properties with specific relation extensions.

Hence, a subset of 17 UD relations (`nsubj`, `csubj`, `obj`, `iobj`, `obl`, `xcomp`, `ccomp`, `amod`, `nmod`, `nummod`, `advmod`, `acl`, `advcl`, `aux`, `cop`, `case`, `mark`) is replaced by 3 major relations in SUD: `subj`, `comp`, `mod` (subject, complement, modifier) with possible sub-relations.[3]

3 SUD relations

SUD relations are organized in a taxonomic hierarchy (Figure 1): A relation that is the daughter of another one inherits its syntactic properties with the addition of specific properties. Indeed, sometimes, we cannot take into account all possible distinctions, either because of the conversion from different treebanks not containing enough information, or because a sentence does not allow to make a clear decision. In those cases, we need a more general class of relations. For example in *They work* `>udep` *at the university* out of context does not allow distinguishing between `mod` and `comp:obl`, and we can then use `udep` (underspecified dependency), the hypernym of `mod` and `comp`. The root of our taxonomy is the `unk` (unknown) relation.

Some UD relations are used in SUD with the same scope and meaning as in UD (in the green frame on Figure 1, whereas UD relations that are not used in SUD are listed in the orange frame), except for some cases where UD is particularly restrictive (see Section 3.2). Also, the sets of `POS` and morpho-syntactic features are similar in SUD and UD.

[2]Moreover, UD is not consistent about when to distinguish clausal complements and modifiers (*He wants to run* is `xcomp` and *He came without running* is `advcl`), while not making the same distinction for adpositional phrases (*He spoke to her* and *He came without her* are both `obl`).

[3]The distinction between arguments and modifiers mainly involve a semantic criterion: an argument of a lexical unit **L** is an obligatory participant in the semantic description of **L** (Mel'čuk, 1988). Although semantic, we want to keep this distinction in the syntactic annotation because most languages have special constructions for arguments such as the English dative shift and the French indirect object complement, which can be pronominalized by a dative clitic.

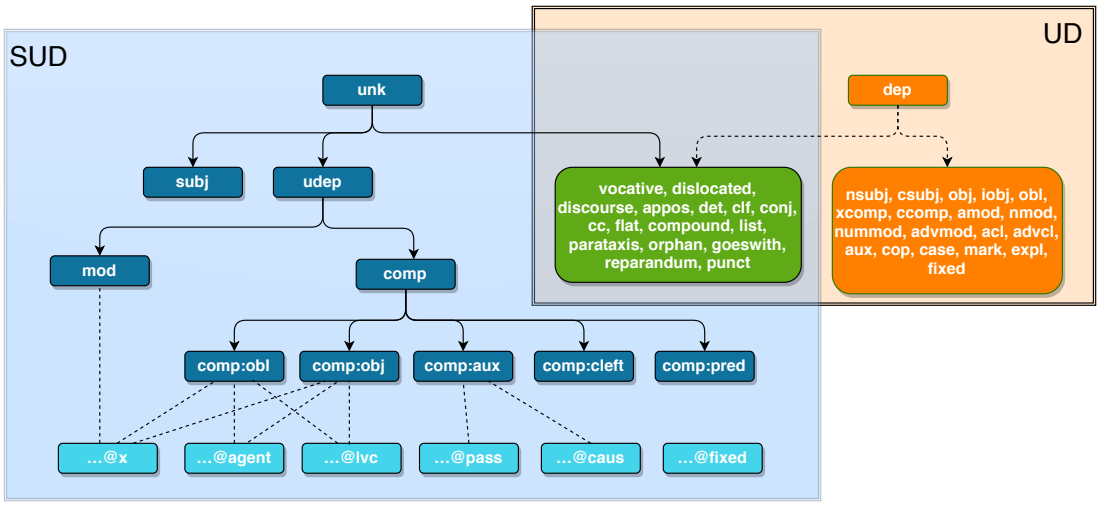

Figure 1: Taxonomy of SUD relations

Figure 2: Cleft sentences

3.1 Surface-syntactic relations

SUD has a unique subject relation, `subj`, and a unique relation `mod` for all modifiers. We will focus here on subrelations of the relation `comp`, for (subcategorized) complements.

- `comp:obj` is used for direct object complements (see examples in the previous section), including direct complements of an adposition or a subordinating conjunction: *about* `>comp:obj` *her*, *whether* `>comp:obj` *(she) leaves*.
- `comp:obl` is used for oblique complements, including clausal complements commuting with an adpositional complement (*I am afraid* `>comp:obl` *of your departure/to leave/that you leave*).
- `comp:pred` is used for predicative complements: *she is* `>comp:pred` *happy*; *she seems* `>comp:pred` *happy*; *I consider (her)* `>comp:pred` *happy*.
- `comp:aux` is used for the complement of an auxiliary: *she is* `>comp:aux` *sleeping*; *she has* `>comp:aux` *left*; (Fr) *elle fait* `>comp:aux` *dormir les enfants* [‘she makes the kids sleep’].
- `comp:cleft` is used for cleft clauses. In Figure 2, the first sentence resembles a relative clause more closely whereas the second sentence is impossible as a relative. Yet, both `comp:cleft` relations depend on *is*.

Due to the functional definition of SUD relations, the span of some UD relations is extended in SUD:

- `det` can be used with numerals in SUD, while all numerals must be `nummod` in UD. To retain the reversibility UD-SUD in the case of a numeral that functions as a determiner, we add a new UD subrelation `nummod:det`.
- Similarly, `discourse` can be used with verbs in SUD, while `parataxis` must be used in UD. In this case, the SUD `discourse` with a verbal dependent becomes `parataxis:discourse` in UD.

3.2 Deep-syntactic features

As explained in Section 2.2, we may have a predicate which does not have its own subject at the surface syntax level. The link of such a predicate with its semantic subject does not concern the surface

syntax but the syntax-semantics interface or what (Mel'čuk, 1988) calls the deep syntax. We decide to explicitely indicate this deep nature by introducing deep-syntactic features on dependencies with the `@` symbol. In the two sentences *He wants to run* and *He came without running*, we introduce a feature `@x`: *wants* `>comp:obj@x` *to* `>comp:obj` *run*, *came* `>mod@x` *without* `>comp:obj` *running*.[4] In other words, `comp:obj@x` is a `comp:obj` surface-syntactic relation whose verbal dependent has its deep subject somewhere in the sentence.[5] This feature, which indicates that the dependent of the relation is not linked to its deep subject, is automatically subsumed by `comp:pred` and `comp:aux` and can be left out for these relations.

This strict separation between surface-syntactic relations and deep-syntactic features is extended to the conversion of other UD relations. For instance, a redistribution (diathesis change) can be signaled as follows:

- `@pass` indicates a passive construction (*she is* `>comp:aux@pass` *fascinated by his attitude*). It can also be borne by the `subj` relation when there is no auxiliary (for example *This business failed miserably, with many of the books* `<subj@pass` *sold as waste paper.*[6]).
- `@caus` indicates a causative construction: (Fr) *il fait* `>comp:aux@caus` *pleurer les enfants* ['He made the kids cry'].
- `@agent` is used for a demoted subject: *she is fascinated* `>comp:obl@agent` *by his attitude*; (Fr) *il fait pleurer* `>comp:obj@agent` *(les) enfants* ['he makes the kids cry'].

UD marks expletive elements with a dedicated relation `expl`. We consider that this is not a surface-syntactic relation, but it is possible to keep this information in the dedicated deep-syntactic feature `@expl`. See an example of an expletive subject in Figure 3.

Note that our annotation scheme remains centered around a surface syntactic analysis, but we isolate semantically-oriented features more explicitly. This allows for an easier interface with the Enhanced UD annotation effort (Schuster and Manning, 2016).

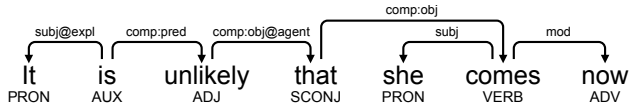

Figure 3: SUD analysis for *It is unlikely that she comes now*

Another example of deep-syntactic features is given by the annotation of light verb constructions: We use the `@lvc` deep-syntactic feature. It is a feature indicating that the dependent is a predicative noun and that the governor is a light verb without semantic contribution. Nouns in light verb constructions can have a `comp:obl@x` dependent.

- (Fr) *Avoir envie de manger* ['having the urge to eat']: *avoir* `>comp:obj@lvc` *envie* `>comp:obl@x` *de* `>comp:obj` *manger*;
- (Fr) *Avoir l'air heureuse* ['having a happy appearance']: *avoir* `>comp:obj@lvc` *air* `>comp:pred@x` *heureuse*;
- (Fr) *Mettre au défi de partir* ['take on the challenge to leave']: *mettre* `>comp:obl@lvc` *à* `>comp:obj` *défi* `>comp:obl@x` *de* `>comp:obj` *partir*.

4 Multi-words expressions in SUD

According to UD guidelines, a special relation `fixed` must be used to annotate some MWE: fixed grammaticized expressions that behave like function words or short adverbials. This notion of fixed expressions tries to take into account two aspects: the fact that there is no clear internal sytactic structure and

[4] We choose `@x` in reference to `xcomp`. We could use `@y` in case of the raising of an object as in *a book easy* `>mod@y` *to read.*

[5] Deep subjects are the first semantic argument of a verb or an adjective. They are labeled as subject in the enhanced UD graph, which is similar to the Mel'čukian deep-syntactic structure.

[6] https://en.m.wikipedia.org/wiki/Honoré_de_Balzac

the fact that the whole expression may have a POS which is not predictable from the POS of the internal tokens (Kahane et al., 2018). We would like to argue that these two aspects are not necessarily linked. In the sentence *He bought heaven knows what*, the idiomatic part *heaven knows what* has at the same time a clear internal syntactic structure and an unexpected POS in the context. SUD recommends an internal analysis of MWEs as soon as there are regular syntactic relations.

To take this into account, we propose to explicitly annotate the POS of a given expression when it is different from the POS of the head token. We propose in SUD to introduce the feature `ExtPOS` (for external POS) to give the POS of the whole expression.

In parallel, we also want to clearly indicated the span of the MWE; this must be done in the deep-syntactic layer because we can have a regular syntactic structure. In such cases, the span of the MWE is indicated by the deep-syntactic feature `@fixed`, added to the relation name (see Figure 4).

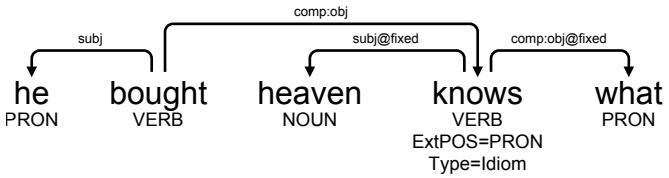

Figure 4: SUD analysis of an idiomatic construction

An alternative to this encoding is the token-based feature method applied in the PARSEME project (Savary and al., 2018; Kahane et al., 2018).

For phrases with no clear internal structure, we indicate at the surface-syntactic level (`unk`) the fact that the relation is unknown and at the deep-syntactic level (`@fixed`) that there is a fixed expression: *each* `>unk@fixed` *other*, *ad* `>unk@fixed` *hoc*.

It is interesting to observe that the fact that some phrase does not behave according to the POS of its head exists also in other contexts not related to MWEs. We also recommend the use of the `ExtPOS` feature in these cases, together with a `Type` feature to explicit the construction:

- In titles (of books, movies, songs…), the head can have various POS but it is most of the times used as a proper noun: *the movie Gone with the wind*, `ExtPOS=PROPN`, `Type=Title`.
- In grafts (Deulofeu, 1999; Deulofeu et al., 2010), which is a phenomenon mainly observed in spoken production, where a clause is used instead of a noun phrase: *he bought I think it is called dowels*, `ExtPOS=NOUN`, `Type=Graft`.

We also suggest that UD should adopt the `ExpPOS` feature or an equivalent mecanism. It will allow for easier generalizations and for more precise validation of the UD treebanks. For instance, in the current validation script of UD, the dependent `advmod` must be `ADV` unless it is a MWE, which means in UD, that the dependent has a `fixed` relation with one of its dependent. If UD adopted a feature-based encoding of MWEs, this condition could be replaced by the presence of `ExtPOS=ADV`, as in SUD.

5 UD ⇔ SUD conversion

The conversion SUD ⇒ UD is done in three main steps (Gerdes et al., 2018): 1) transforming the string analysis for the relation `conj` into a bouquet structure; 2) reversing relations `comp:aux`, `comp:pred` with an `AUX` governor, and `comp:obj` with an `ADP`, a `SCONJ`, or a `PART` governor (which gives us `aux`, `cop`, `mark` and `case`); 3) mapping other SUD relations directly to UD relations.

Two types of extensions in relations are considered:

- Some extensions are associated to special rules. For instance, `comp:pred` gives us `xcomp` (when the governor is not an `AUX`), as well as `comp:obj@x` and `comp:obl@x`.
- Deep-syntactic extensions are just copied as simple extensions, because the notion of deep-syntactic extension does not exist in UD. For instance, `comp:aux@caus`, gives us `aux:caus`.

In the UD ⇒ SUD conversion, the three same steps are applied.[7] Extensions used in UD which are unknown in SUD are just copied but with the symbol `@`. For instance, `case:loc` used in different Chinese UD gives us `comp:obj@loc` in Chinese SUD.[8]

In order to avoid confusion between SUD and UD, relations which are common to the two annotation schemes have the same interpretation in both schemes. In some marginal cases (`det` or `discourse`), we allow a wider use for a relation in SUD.

It must be noted that there is not a one-to-one correspondence between SUD and UD relations, because the relations are defined on different principles. Nevertheless, in most cases, the conversion is reversible. For instance, UD `xcomp` corresponds to `comp:pred`, `comp:obj@x` and `comp:obl@x`, but in general the relation can be recovered according to the dependent's POS (`ADJ` or `NOUN` for `comp:pred`, `VERB` with or without a marker in the other cases).[9] Conversely, the SUD `mod` relation corresponds to several UD relations according to the POS of the governor and the dependent (`amod`, `nummod`, `nmod`, `acl`, when the governor is a `NOUN`; `advmod`, `advcl`, `obl:mod` when the governor is a `VERB`). In case the `ExtPOS` feature is instantiated, it must be used for the determination of the UD relation, and not the regular `POS` feature.

6 Conclusion

The SUD principles have been further refined in this article:

- SUD must be translatable in UD, but SUD can be more precise than UD (cf. the case of UD `xcomp`).
- SUD tries to make a clear distinction between surface-syntax properties, only based on distributional criteria, and deep-syntactic properties, concerning the syntax-semantics interface.
- SUD needs an encoding of the `POS` of MWEs, since this is no longer encoded in the relation name.

SUD is available for the development of new treebanks. A github project dedicated to SUD is under construction at https://surfacesyntacticud.github.io/, which will collect all available resources for SUD: universal and language-specific annotation guidelines, natively annotated SUD treebanks, SUD treebanks automatically converted from UD, GREW grammars (Bonfante et al., 2018) for the conversion UD ⇒ SUD and SUD ⇒ UD, and other consolidation tools.

We hope that this alternative annotation scheme opens up the world of UD to communities that have been reluctant to adopt some UD annotation choices. Moreover, SUD is not only a well-grounded and validated annotation scheme that has been successively applied to languages of various language groups, the conversion tools and practice that we propose are designed for an easy deployment to other alternative annotation schemes around the UD project.

References

Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*, volume 1 of *Logic, Linguistics and Computer Science Set*. ISTE Wiley.

Bernard Caron, Marine Courtin, Kim Gerdes, and Sylvain Kahane. 2019. A surface-syntactic ud treebank for naija. In *Proceedings of Universal Dependencies Workshop*, Paris.

Marine Courtin, Bernard Caron, Kim Gerdes, and Sylvain Kahane. 2018. Establishing a language by annotating a corpus: The case ofnaija, a post-creole spoken in nigeria. In *Proceedings of the workshop on Annotation in Digital Humanities (An-nDH)*, pages 7–11, Sofia.

[7] In this case, the second step is more delicate, because some elements must be raised to the functional head. For instance, in English, the negation is clearly borne by the auxiliary (*she has not slept*) because an auxiliary must always be present in case of negation (*she does not sleep*).

[8] In the conversion evaluated by (Gerdes et al., 2018), relations with unknown extensions where not treated, which was the source of many problems of non-reversibility.

[9] Yet, in some specific cases it is not possible to recover the SUD relation corresponding to the UD relation. For instance the `xcomp` relation in French can correspond to two different SUD relations for similar surface forms: *il rêve de venir*['he dreams of coming'] commutes with *il rêve de ça* ['he dreams of that'] and is a `comp:obl@x`, while *il tente de venir* ['he tries to come'] commutes with *il tente ça* ['he tries that'] and is a `comp:obj@x`.

Henri-José Deulofeu, Lucie Dufort, Kim Gerdes, Sylvain Kahane, and Paola Pietrandrea. 2010. Depends on what the french say: Spoken corpus annotation with and beyond syntactic function. In *Proceedings of the 4th Linguistic Annotation Workshop (LAW IV)*, Uppsala, Sweden.

Henri-José Deulofeu. 1999. *Recherches sur les formes de la prédication dans les énoncés assertifs en français contemporain (le cas des énoncés introduits par le morphème que)*. Thèse d'état, Université Paris 3.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2018. SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In *Universal Dependencies Workshop 2018*, Brussels, Belgium, November.

Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Richard A. Hudson. 1984. *Word grammar*. Oxford: Blackwell.

Richard A. Hudson. 1987. Zwicky on heads. *Journal of linguistics*, 23(1):109–132.

Sylvain Kahane, Marine Courtin, and Kim Gerdes. 2018. Multi-word annotation in syntactic treebanks: Propositions for universal dependencies. In *Proceedings of the 16th international conference on Treebanks and Linguistic Theories*, Prague.

Haitao Liu. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*.

Igor Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. Albany, N.Y.: The SUNY Press.

Joakim Nivre and al. 2019. Universal dependencies 2.4. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Adam Przepiórkowski and Agnieszka Patejuk. 2018. Arguments and adjuncts in universal dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3837–3852, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Agata Savary and al. 2018. *PARSEME multilingual corpus of verbal multiword expressions.* Language Science Press.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *LREC*.