Starting a new treebank? Go SUD!
Theoretical and practical benefits of the Surface-Syntactic distributional approach

Kim Gerdes
Lisn, CNRS,
Université Paris-Saclay
gerdes@lisn.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

The paper brings to the fore some advantages to first develop a new treebank in Surface-Syntactic Universal Dependencies (SUD) annotation scheme, even if the goal is to obtain a UD treebank. Theoretical benefits of SUD are presented, as well as UD-compatible SUD innovations. The two-way UD ⇔ SUD conversion is explained, as well as the possibility to customize the conversion for a given language. The paper concludes by a practical guide for the development of a SUD treebank.

1 Introduction

SUD, Surface-Syntactic Universal Dependencies, is a syntactic annotation scheme, which is a convertible variant of Universal Dependencies (UD). UD is a very successful treebank development project that is now an indispensable standard of data-based syntax (de Marneffe et al., 2021). To benefit from UD's wealth of expertise, tools, and cross-language comparability, any annotation scheme must eventually be convertible into UD. Nevertheless, the UD annotation scheme was initially developed in the context of NLP applications, rather than pure linguistic considerations and some initial choices are problematic.[1] SUD is based on a different theoretical framework that has many advantages for treebank development as we will show in this paper.

SUD has already been presented in two papers by (Gerdes et al., 2018; Gerdes et al., 2019). While SUD's theoretical foundations remain unchanged, this paper proposes one change of SUD's philosophy. At first, SUD was thought of as a pure variant of UD with a complete equivalence between SUD and UD. Initially, SUD was more interested in the UD ⇒ SUD conversion because for some studies, especially on word order typology, a more surfacic annotation was required.[2] This paper reports on a growing interest in SUD ⇒ UD conversions and the development of treebanks in SUD in order to obtain both SUD and UD variants of the treebank. The UD ⇒ SUD conversion grammar is still maintained and has even been improved with the possibility to more easily customize the conversion for a given language. Recent views on SUD abandons the idea of having an equivalence between the two annotation schemes, and this

[1] UD is initially based on Stanford dependencies, which was itself the conversion into a dependency tree of the outputs of a phrase-structure-based parser. In consequence, UD dependency relations combine both functional and categorical information, for instance with the nsubj vs csubj distinction between nominal and clausal subjects, the obj vs ccomp distinction between nominal and clausal objects, or the amod vs nmod vs advmod distinction between adjectival, nominal, and adverbial modifiers, as well as the obl vs nmod distinction between adpositional phrases depending on a verb or a noun. Moreover, UD is very semantically-oriented, favoring relations between content words, leaning towards a sort of interlingua representation. The part of speech tags, stemming from Google's universal POS (Petrov et al., 2012) and the Interset interlingua tagset (Zeman, 2008), were added independently, resulting in some redundancy.

[2] Let us recall that in UD function words depend on content words. As a consequence, adpositions are dependents of the noun with which they form a phrase. This is in complete contradiction with typological studies that show that the adposition-noun relation tends to have similar properties than the verb-object relation. In particular, VO languages have prepositions while OV languages have postpositions (Dryer, 1992).

paper postulate that SUD is a richer annotation scheme than UD. In other words, no information is lost in UD ⇒ SUD and a double conversion UD ⇒ SUD ⇒ UD should give the initial treebank, eventually with additional features.[3] But a SUD ⇒ UD conversion generally causes a loss of information and SUD treebanks obtained from a UD conversion are underspecified for some features considered as relevant for the SUD annotation scheme, such as the internal structure of nuclei or of MWEs. As a simple example consider the verbal chain in the sentence *I would have left*. SUD annotates the hierarchical relation between the three verbs (*would* → *have* → *left*), UD sees a flat structure in these three verbs with the lexical verb (*left*) at its head. Therefore, the hierarchical relation between *would* and *have* is not encoded in UD, and requires language specific heuristics to obtain the correct SUD structure. Theoretical benefits of SUD are presented in Section 2 and completed in Section 3 by UD-compatible SUD innovations.

Due to the fact the SUD is richer than UD, we encourage developers of treebank to start with a SUD annotation, which allows them to obtain a high-quality UD treebank, while keeping information that is flattened out in UD. Moreover if a treebank already exists in a third format, it can be easier to convert it into SUD and only then into UD rather than to aim UD directly because of the unconventional lexical-word-centric approach of UD. We may further assume that SUD's additional richness does not slow down the overall annotation process as it also removes some redundancies of UD. The UD ⇒ SUD and SUD ⇒ UD conversions are presented in Sections 4 and 5, as well as the possibility to customize the conversions for a given language. Section 6 sketches a practical guide for the development of a SUD treebank.

2 Theoretical benefit of SUD

We discuss four benefits of SUD compared to UD: a definition of dependency based on distributional criteria, an encoding of the internal structure of nuclei, a definition of syntactic relations based on commutation positional paradigms, and a more symmetrical analysis of coordination. These properties are core elements that cannot be integrated in UD, which is based on different fundamentals. Other benefits of the current SUD annotation that could be adopted in UD are presented in Section 3.

2.1 Definition of dependency based on distributional criteria

UD favors relations between content words, while function words are treated as dependents of content words. While it may seem at first view that it is easy to establish the difference between function and lexical words for a new language, it turns out to be a hard task to delimit the content word - function word opposition that is compatible with a coherent non-catastrophic annotation.[4] Moreover, supposing that the opposition is semantic or language independent can lead to erasing typologically important structural differences, for example when languages differ precisely in the structure of function words. Relegating all function words as done by UD makes us loose some syntactic information as we will see in the next section.

SUD favors a definition of the dependency structure based on a more traditional definition of head: The head of a unit U is the element A that controls the distribution of U. By *distribution*, we mean what Mel'čuk's (1988) calls the *passive valency*, that is, the set of possible syntactic governors for U, or, similarly, the set of syntactic positions that U can occupy. Even if the notion of governor is based on the notion of distribution, we avoid the circularity, because in most cases the question of the head is not controversial, especially for the governor of a sentence.

As soon as we can determine units and a head for each unit, we have a dependency structure (Gerdes and Kahane, 2013): B depends on A as soon as A is the head of the unit that A and B form together.

This definition of the head is based on formal criteria that we want to recall here because they have often been misstated. Let us consider a unit U = AB. The simplest case is when A or B can stand alone.

[3] The lossless conversion might require language-specific rules, see Section 4.

[4] We use *catastrophe* here in a strictly mathematical sense of Thom's catastrophe theory (Saunders, 1980), i.e. a brutal structural change in a continuum. In the case of annotation, this boils down to very similar constructions ending up with very different syntactic structures, see (Gerdes and Kahane, 2016) for details.

In this case the distribution of A or B can be considered and compared with the whole unit U.[5] It gives us two criteria.

Positive distributional criterion with deletion. If U = AB, A can stand alone (i.e., B can be deleted), and U and A have the same distribution, then A is a head of U.

Negative distributional criterion with deletion. If U = AB, B can stand alone, and U and B do not have the same distribution, then A is a head of U.

The second criteria can be applied to examples such as U = *John ran* or U' = *with John*, where B = *John*. Clearly B does not have the same distribution as the clause U or the phrase U' and then the verb is the head of U and the adposition the head of U'. In the same way, a combination auxiliary-verb such as U = *is expected* has the auxiliary as head, because the past participle has a different distribution: It can be the dependent of a noun (*that's the guy expected at noon*), while *is expected* can be the dependent of a verb (*he knows he is expected*).[6]

It is not needed to delete an element to decide which element is the head, a commutation with another element is sufficient:

Distributional criterion without deletion. If U = AB, A can commute with an A', and U and U' = A'B does not have the same distribution, then A is a head of U.[7] In other words, if B depends on A, then B must not modify the distribution of A and a commutation on B does not change the distribution of the unit it forms with A.

For instance, U = *with John* and U' = *by John* have different distributions. In other words, the commutation of *with* and *by* change the distribution, which implies that the preposition is the head. The same criteria can be used with the determiner-noun combination: Some nouns such as *day* (*she stayed two days*) or *time* (*I will do that (the) next time*) have a very special distribution, being able to work as an adverbial phrase, whatever the determiner is. This is a good argument to take the noun as the head, even if there are also arguments to take the determiner as the head.

2.2 Internal structure of nuclei

In a recent paper on UD, de Marneffe et al. (2021) justify treating function words as dependents as follows: "Sometimes linguistic head functions are divided between a structural center (an auxiliary or function word) and a semantic center (a lexical or content word), such as for periphrastic verb tenses like *has arrived*. This is what Tesnière (2015 [1959], ch. 23) refers to as a dissociated nucleus. In such cases, UD chooses the lexical or content word as the head, and makes function words dependents of the head in the dependency tree structure, while recognizing that they do form a nucleus together with the content word." Nevertheless in case of the presence of multiple function words, Tesnière considers that there is an embedding of nuclei, while UD only considers a flat structure with all function words depending on the same content word and the internal structure of the nucleus is completely lost. For instance, in the sentence of Fig. 1, *in Mesoamerica* is clearly a nucleus that is put in a comparison with *in the Americas* and then embedded in *than in Mesoamerica*.[8] The UD analysis does not have a phrase *in Mesoamerica*.

In the UD ⇒ SUD conversion, we use heuristics that are described in Section 4, depending on the order of the function words and their function. In particular, the closer a function word is to the content word, the earlier they combine. The SUD structure of the same sentence is given in Fig. 1 (lower part).

[5] When comparing the distribution of two units, we mainly use our intuition. For tricky cases, we also observe the actual distribution in our corpora, but nothing is completely currently formalized.

[6] *is expected* can also be the dependent of a noun, but only if it combine with a relativizer (*the guy that is expected at noon*) and, in this case, it is the relativizer that is head of the relative clause, because the relativizer change the distribution of *is expected*.

[7] When saying that A' can commute with A, we are only considering the commutation in the context of B. In other words, this means that A'B is a valid combination and that A and A' exclude each other in this context (i.e. AA'B is not valid).

[8] Note that the analysis of comparative complements is erroneous in English UD treebanks: *than in Mesoamerica* should depend on *more* and not on *obvious*, because *more than in Mesoamerica* is a valid sub-unit of the sentence and not **obvious than in Mesoamerica*.

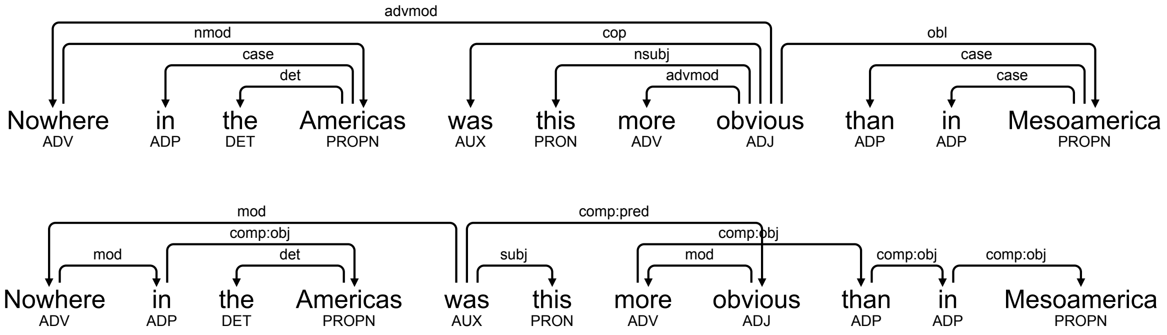

Figure 1: UD and SUD analysis of Sentence *Nowhere in the Americas was this more obvious than in Mesoamerica.* (GUM_textbook_history-19)

But this heuristic does not work in some cases. For instance, Wolof has a multitude of auxiliaries that are used to focus the subject, a complement, or the verb itself, which will occupy the first place in the clause (Robert, 1991; Bondéelle and Kahane, 2021). The auxiliary *na*, used to focus a verb, can also focus an auxiliary, as in Fig. 2 where the past imperfective auxiliary *doon* is focalized by *na*, which is the head of the nucleus *doon na* VERB. Here, *na* is the closest function word to the content word, but it combines last.

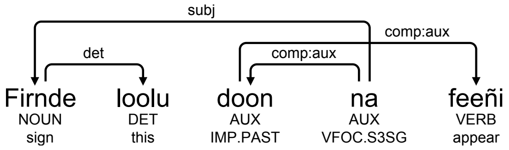

Figure 2: SUD analysis of the Wolof sentence *Firnde loolu doon na feeñi* 'This sign was to be revealed.'

Another problematic case is when there are function words on both sides of the content word. This can be illustrated by the auxiliaries in German, as in sentence (1).

(1)	*Jeder*	*siebte*	*Beschäftigte*	*wird*	*dann*	*seine*	*Kündigung*	*erhalten*	*haben*
	Each	seventh	employee	will	then	his	notice	received	have

'One in seven employees will have received their notice by then.'

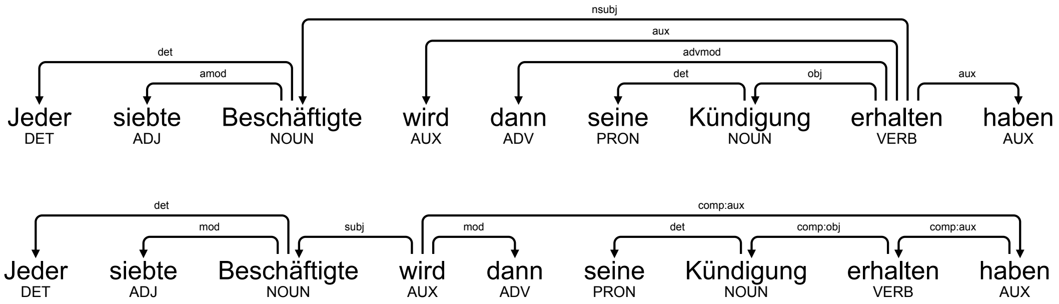

Figure 3: UD and SUD analysis of Sentence (1)

German is a V2 language, where the finite verbal form always occupies the second position of a declarative sentence, whether it is a content verb or an auxiliary. In (1), the verb has two auxiliaries, *wird* 'will' on the left and *haben* 'have' on the right. The auxiliary on the left, which is in the second position in the sentence and has a finite form, is the root of the syntactic structure, which cannot be guessed from the flat UD structure alone.

2.3 Definition of syntactic relation based on positional paradigms

In SUD, two dependents that belong to the same positional paradigm have the same syntactic relations, in accordance with Mel'čuk's (1988) or Van den Eynde & Mertens' (2003) definitions, while UD takes also into account the POS of the governor and/or the dependent (see Note 1 about the definition of relations in UD). One advantage of the SUD definition is the possibility to compare the valency of two occurrences of the same lemma and to extract a syntactic lexicon more easily.

As UD, SUD uses the notation `rel:subrel` for a sub-relation of a given relation. Syntactic relations are part of a hierarchy and `comp:obj` or `comp:obl` must be understood as sub-relations of a more generic `comp` relation. Modifiers (`mod`) and complements (`comp`) are distinguished, but a super-relation `udep` (underspecified dependency) can be used if we do not want to make this distinction. We use it for noun dependents and it is used in non-native SUD treebanks for the conversion of the UD `obl` relation, which gives the `udep` relation in SUD.[9] Figures 4, 5, and 6 give UD and SUD annotations of verb dependents which are respectively modifier, argument and underspecified. Annotations in Figures 4 and 5 are SUD-native and contain a distinction between complements and modifiers, which is kept in the conversion with the UD relations `obl:arg`, `iobj`, and `obl:mod`. Conversely, the sentence in Figure 6 comes from UD_ENGLISH-GUM, where the distinction between complements and modifiers is not present for preposition phrases and the conversion to SUD gives us a `udep` relation.

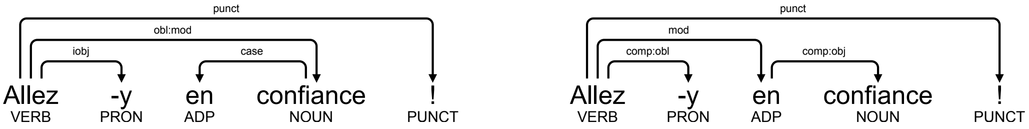

Figure 4: UD and SUD analysis of *Allez-y en confiance !* 'Go there with confidence!'

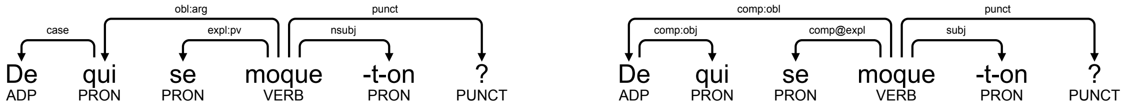

Figure 5: UD and SUD analysis of *De qui se moque-t-on ?* 'Who are we kidding?'

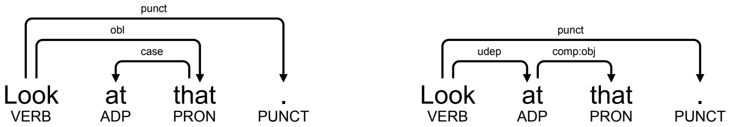

Figure 6: UD and SUD analysis of *Look at that.*

Additional features on relations are clearly separated from the relation itself, especially when it is semantic information. We use for this the delimiter `@`. For instance, the semantic value of an auxiliary (tense, passive, causative) can be indicated on the `comp:aux` relation: `comp:aux@tense`, `comp:aux@pass`, `comp:aux@caus`. Subjects all have the function `subj`, but expletive or passive subjects can be marked by an additional feature: `subj@expl`, `subj@pass`.[10] In spoken corpora, the feature `@scrap` has been used for incomplete units. This feature is particularly useful for error mining:

[9] UD uses the `obl` relation for all adpositional phrases depending on a verb, but for clauses depending on a verb, a distinction is made between complements (`ccomp` or `xcomp`) and modifiers (`advcl` for adverbial clauses).

[10] Contrary to UD, SUD does not have an `expl` relation for expletives. We consider that *it* in *it is impossible to do that*, is above all a normal subject and is analysed as `subj@expl`.

for instance, a relation between a verb and a determiner (in an incomplete sentence such as *I see the...*) should not be allowed without a `@scrap`.

2.4 A more symmetrical analysis of coordination

In UD, the dependent shared by all the conjuncts are attached to the head of the coordination, the leftmost conjunct.

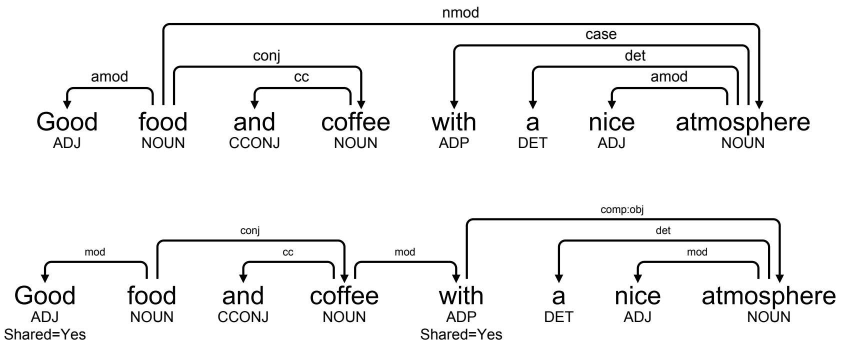

Figure 7: UD and SUD annotation of *Good food and coffee with a nice atmosphere*

In the example of Fig. 7, from the UD_ENGLISH-EWT corpus, there are two modifiers of the coordination *food and coffee*: a left modifier *Good* and a right modifier *with a nice atmosphere*. Since the right modifier is after the second conjunct, the UD annotation has only one interpretation: It cannot be the modifier of the first conjunct alone but only of the coordination as a whole. However, for the left modifier, the UD annotation does not indicate whether it is a modifier of *food* only or of *food and coffee*. This is an unfortunate asymmetry.

In SUD, as in UD, the head of the coordination is the head of the leftmost conjunct, but for the dependents, the annotation is perfectly symmetrical. They are attached to the nearest conjunct: the left to the leftmost conjunct and the right to the rightmost conjunct. In order to indicate which dependents are shared, we introduce the feature `Shared` with values `Yes` and `No`. Conversions of UD treebanks, only give a partial instantiation of the `Shared` feature. In the native SUD_FRENCH-GSD, `Shared=Yes` features have been systematically introduced. Note also the considerably shorter overall dependency lengths of the SUD annotation scheme, which is not only cognitively more plausible but also facilitates manual annotation and correction.

3 UD-compatible SUD innovations

This section presents features of the SUD annotation scheme that could, and we believe should, be integrated into the UD annotation guidelines. For now, the SUD ⇒ UD conversion will encode these SUD features as optional additional information in the MISC column.

3.1 Internal structure of Multi-Word Expressions

Multi-Word expressions (MWE) cover a wide heterogeneous field of constructions such as use of foreign words that have no internal structure in the host language (*Burkina Faso*, *Hong Kong*, *ad hoc*), or completely regular structures in named entities (*the Embassy of Ecuador in London*, *the United States*). Interesting from a syntactic point of view is another set of phenomena: constructions that have a regular internal structure but that intervene as a whole at an unexpected point in the sentence. For example, *in order* (*to* VERB) is analyzed as an MWE in English treebanks, as shown in Fig. 8 (upper part) from UD_ENGLISH-GUM, with a `fixed` relation between *in* and *order*.

Even if *in order* is semantically frozen it is nevertheless a syntactically regular preposition-noun combination. In native SUD, the sentence is analyzed with the standard `comp:obj` relation between *in* and *order* (the noun is the object of the adposition) and the idiomaticity is encoded by additional features `Idiom=Yes` on the head and `InIdiom=Yes` on the other elements.

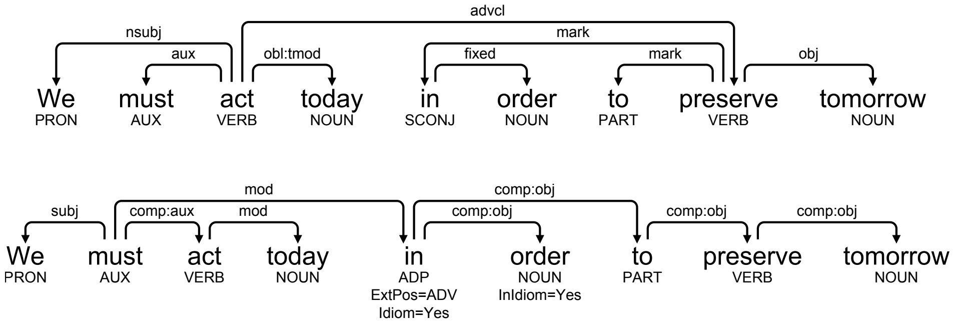

Figure 8: UD and SUD annotation of Multi-Word Expressions

Moreover, we consider that *in order* as a whole works as an adverb, which is encoded in SUD by the feature `ExtPos=ADV` (for external POS).[11] Of course, this SUD analysis translates into a different UD analysis, because adverbs are analyzed as content words.[12] Arguably, the UD analysis would have been different if the internal structure of the MWE had been taken into account.

3.2 Textform and wordform

It was identified in UD that, in several places, syntactic units do not exactly correspond to orthographic units given in the raw text.[13] For instance, in French the orthographic unit *au* is a contraction of two syntactic units: the preposition *à* and the determiner *le* (such amalgams are called Multi-Word Tokens or MWT). With a focus on syntax, it is natural to consider syntactic units as the basic units of annotation; this is what is done both in UD and in SUD. However, it is necessary to keep all the information and to also encode the orthographic unit when it differs from the units of the structure. The UD guidelines[14] introduce the CoNLL-U format with a dedicated mechanism with a new type of line describing a range of tokens (`2-3` in the example below) to store the contracted form.

```
2-3 au _ _
2 à à ADP
3 le le DET
```

The main drawback of this solution is that the syntactic dependency structure, being based on the syntactic units, does not refer to the orthographic units which are then not easily accessible for tools working on the syntactic structure. Having access to these orthographic units is useful for parsing.

There are other cases where an orthographic unit is different from a canonical token. For instance, for several languages, uppercase letters are used at the beginning of a sentence, in specific usages for naming institutions (*the White House*), in titles (*What the Moon Brings* [GUM_fiction_moon-1]), or for emphasis (*YES!*). It is useful to encode the canonical form in these cases, as it allows for an improved data analysis, performing linguistic queries on canonical forms.[15]

We propose a new way to encode the orthographic information in these two cases (MWTs and non-canonical forms) with two new features: `textform`, which always contains orthographic data and `wordform` which always contains a canonical lexical form (see Table 1 for examples).

[11] English has adverbs taking a *to* VERB complement, such as *up*, *next*, *about*, or *prior*, but there are no subordinating conjunctions with this valency.

[12] In order to keep a function word status for *in order*, *in* has been analyzed in the UD analysis of Fig. 8 as a subordinating conjunction (`SCONJ`, as in all occurrences of *in order* in UD_ENGLISH-GUM, version 2.8), which is surprising to say the least.

[13] Here, *orthographic* means the actually observed letters in input text.

[14] `https://universaldependencies.org/format.html#words-tokens-and-empty-nodes`

[15] Note that this canonical form may not be trivial to recover. In French, diacritics are optional on upper-case letters, and an *A* as the first word can be either the preposition *à* (ex: *à qui tu penses ?* 'who are you thinking of?') or a verbal form *a* (*a-t-il choisi ?* 'has he chosen?').

	`form`	`lemma`	`textform`	`wordform`	`CorrectForm`
[fr] au	à	à	au	[à]	
	le	le	_	[le]	
[en] wanna	want	want	wanna	[want]	
	to	to	_	[to]	
[en] The	The	the	[The]	the	
[fr] Le maison	Le	le	[Le]	le	La
[en] egg plant	egg	egg	[egg]	eggplant	
	plant	plant	[plant]	_	
[en] NEEEVERR	NEEEVERR	never	[NEEEVERR]	neeeverr	never

Table 1: Examples on the usage of features `textform`, `wordform` and `CorrectForm`.

The main advantage is that, using features, all information is available in the units used in the syntactic structure and it makes it possible to use these features in any tool (for querying the treebank, for conversion. . .).

It might seem appealing to use these features for encoding typos as well. But, there may be conflicts, as shown for the phrase [fr] *Le maison*: *Le* must be corrected in *La* (the gender of *maison* is feminine) but also be normalised into *le*. So, we decided to use the feature `CorrectForm` (already used in other UD treebanks) in case of typos, to express the way it should be written.

In order to avoid having an overly verbose CoNLL file, we propose in practice, to explicitly record `textform` and `wordform` only when they are different from the feature `form` (column 2 in CoNLL). In Table 1, square brackets are used to show feature values which are not stored in the CoNLL file.

4 The conversion UD ⇒ SUD

Our approach of the conversion between different syntactic annotations is based on graph rewriting. Each annotation is seen as a graph and the conversion of an annotation into another annotation is performed by applying a sequence of local graph rewriting rules. For this, we use the GREW tool[16]. In Grew, a Grew Rewriting System (GRS) is a set of rewriting rules organized into strategies such that these rules can be ordered, iterated and grouped into packages.[17]

Since SUD is richer than UD, a universal UD ⇒ SUD GRS can only approximate the correct SUD annotation due to the lack of information in the UD annotation, and the adaptation of the GRS to each language is crucial.

4.1 The universal conversion UD ⇒ SUD

The universal UD ⇒ SUD system has five main tasks to perform:

1. Replacing UD dependency labels with SUD dependency labels.
2. Reversing some dependencies between function words and lexical words to change the heads of adpositional phrases, subordinate clauses, and verb-auxiliary pairs.
3. Shifting the source of some dependencies as the result of reversing dependencies.
4. Attaching the right dependents of coordinations to the rightmost conjunct, whereas in UD they are attached to the leftmost conjunct, the coordination head (see Section 2.4).
5. Transforming bouquets of coordinated elements into sequences, marking embedded coordinations with the `emb` extension added to `conj` relations.

[16] `https://grew.fr/`

[17] All GRS described in this section are available on `https://github.com/surfacesyntacticud/tools/tree/master/converter`

These tasks are not independent of each other and although they can most often be carried out in any order, their forms depend on this order and sometimes one order is more relevant than another. The universal UD ⇒ SUD GRS contains 89 rules grouped into 20 packages.

As said above, a conversion of an UD annotation into a SUD annotation is necessarily approximate. The lack of information is particularly problematic in four cases:

1. when several function words depend on the same lexical word in UD (see Section 2.2),
2. when a UD dependency from a lexical word to a function word has to be reversed, some of its dependents has to be transferred to the function word but there is usually no indication on which dependents have to be transferred,
3. to decide whether left dependents of a coordination head are dependent of the whole coordination or of the head alone,
4. when idioms have an internal structure, which is not represented in UD and cannot be recovered in the conversion.

For the first problem, we assume that the further a function word is from the content word, the higher it is in the dependency structure, but there are cases that cannot be solved by such an heuristic, as shown with auxiliaries in Wolof and German (Section 2.2), and our conversion necessarily produces errors without a language-by-language customization.

For the second problem, we have implemented some rules for specific cases: for instance, the subject moves to the auxiliary, while the complements stay on the lexical verb. For modifiers, it is more complex and we resort to word order, preserving the projectivity as much as possible, but only a language-specific and lexicon-based conversion could ensure a perfect structure.

For the third problem, we use heuristics to decide. For example, if the leftmost conjunct of a coordination has a subject to its left and the other conjuncts have no subject, we consider that the subject is shared by all conjuncts.

For the fourth problem, UD flat structures of idioms are converted into SUD flat structures.

4.2 Customization of the UD ⇒ SUD conversion

We have presented default solutions that minimize errors in the UD ⇒ SUD conversion. By customizing the GRS for specific languages, we can further reduce the errors.

For the case of several function words depending on the same lexical word, our architecture allows us to attribute a feature level to dependencies being to reverse with a value that gives its priority in the reversing process. For instance in French, `cop` dependencies are assigned a bigger priority than `aux` dependencies, which means that in case of competition `cop` dependencies must be reversed before `aux` dependencies. Such a rule is needed when the predicate has been extracted as in Fig. 9.

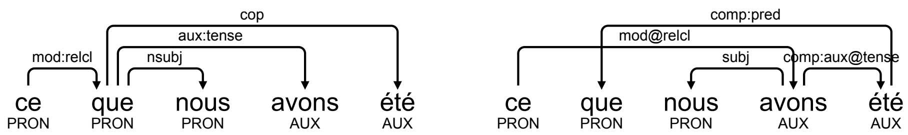

Figure 9: UD (left) and SUD (right) trees for *ce que nous avons été* ‘what we have been’

For the moment, the UD ⇒ SUD conversion has been customized for French and Wolof. For French, a lexicon of modifiers that must move to the auxiliary has been developed. For Wolof, the level mechanism is used to take into account the case described in Section 2.2.

5 The conversion SUD ⇒ UD

Since SUD is richer than UD, we should have no difficulty in designing a universal GRS that converts any SUD annotation of a corpus in any language into an UD annotation. This is globally true but conversion sometimes requires adaptation to the specificity of the language.

The universal SUD ⇒ UD GRS must perform the same tasks as the universal UD ⇒ SUD GRS (see Section 4.1), but in the opposite direction, and the rule order is not the same. It currently contains 94 rules grouped into 20 packages.

In UD, the label of a dependency takes into account not only the syntactic function realized by the dependency but possibly the POS of the governor and the POS of the dependent. For example, the SUD `mod` dependency is converted into a UD `advmod`, `amod`, `nmod`, `obl` or `advcl` dependency, and knowledge of governor's and dependent's POS does not always identify the dependency label. In specific contexts, some words are not used in their usual syntactic function and this use depends on the language.

For example, a SUD `mod` dependency from a verb to a noun is by default a UD `obl` dependency, but there are exceptions. Examples (2) from UD-ENGLISH-GUM illustrate respectively the two cases.

(2) (a) ***Many times** prideful people have a serious 'my-way's-the-only-way' attitude.*
(b) ***An undistinguished student and an unskilled cricketer**, he did represent the school.*

In SUD, the dependencies *have* → *times* (2a) and *represent* → *student* (2b) are both `mod` dependencies. The first one becomes an `obl` dependency in UD, whereas the second one becomes an `advcl` dependency because the noun phrase *an undistinguished student and an unskilled cricketer* is considered as a clause with an ellipsis equivalent to *being an undistinguished student and an unskilled cricketer*.

Since there is no universal criterion to distinguish the two cases, we have designed a SUD ⇒ UD conversion rule, which transforms `mod` relations into `advcl` relations if the governor is a verb and the dependent is a non-temporal nominal preceding the verb, but such a rule only works for certain languages, French and English in particular. Since the rule requires distinguishing temporal nominals, we chose to link the conversion rule to a lexicon. Another solution would have been to mark temporal nominals in the corpus (as it is done in some treebanks with the `tmod` extension).

Another difficulty in the SUD ⇒ UD conversion is that the definition of some UD relations takes into account semantic properties. In particular, the relation between a verb and an argument clause is denoted `xcomp` if the subject of the object clause is controlled by the main verb. Otherwise, the relation is denoted `ccomp`. Consider the following examples extracted from the FRENCH-GSD corpus.

(3) (a) *les mesures visant à développer l'accord* 'measures (aiming) to develop the agreement'
(b) *Le tourisme commence à se développer.* 'Tourism is starting to develop.'

The UD annotation of (3a) includes a *visant* −[`ccomp`]→ *développer* dependency, whereas the UD annotation of (3b) includes a *commence* −[`xcomp`]→ *développer* dependency. In SUD, both dependencies are denoted `comp:obl` according to the fact that the definition of syntactic relations is based on positional paradigms (see Section 2.1). To choose between `xcomp` and `ccomp` in the SUD ⇒ UD conversion of these relations, a way is to use a lexicon of control verbs and a conversion rule, which uses this lexicon. A major drawback is that you it should be done for each language separately. To avoid this drawback, another way is to mark the relations of the control verbs to the concerned argument with a special feature. That is what is done with the extension `@x` in the SUD annotation.

The method we just described for improving the UD annotation resulting from the conversion can be used to take into account the idiosyncrasies of some languages. The diverse interests behind treebank development regularly lead to some idiosyncratic enrichment of the annotation. UD responds to this need with the option of adding language (or treebank) specific subrelations and features, and SUD naturally follows this approach. If and only if the SUD treebank developers have added new subrelations or features and want them to be taken into account when translating to UD, they must add these idiosyncratic rules to the universal SUD ⇒ UD GRS.

For the time being, the SUD ⇒ UD conversion has been customized for French (by inserting two rule packages in the universal GRS), Naija, and Beja. For Beja, which is a strongly head-final language, coordinations have been analyzed in SUD by head-final `conj` relations (see (Kanayama et al., 2018) for a similar analysis in Japanese and Korean). As `conj` relations must always be head-initial in UD, we have added an ad hoc conversion to a `dep:conj` relation, but it is possible to customize the conversion in another way, for instance, by reversing the direction of `conj` relations.

On the train part of SUD_FRENCH-GSD, the language-specific customization fixes 1.2% of the 400,220 dependencies in the UD ⇒ SUD direction and 0.4% in the other direction (i.e threes times less, which is not surprising). The low percentage shows that idiosyncratic customization can be ignored at first when starting a SUD treebank as the universal SUD ⇒ UD conversion amply does the trick.

The lack of gold annotation in UD and SUD does not allow a direct evaluation of our SUD ⇒ UD and UD ⇒ SUD conversion tools, but we have done an indirect evaluation, using double conversion. The SUD ⇒ UD conversion followed by the UD ⇒ SUD conversion on the SUD_FRENCH-GSD corpus gives 6231 different dependencies out of 400,220 dependencies, i.e. 1.56% of the total, between the resulting annotation and the initial annotation. The UD ⇒ SUD conversion followed by the SUD ⇒ UD conversion on the UD_FRENCH-GSD corpus gives 90 different dependencies out of 400,220 dependencies, i.e. 0.02% of the total, between the resulting annotation and the initial annotation. This highlights that SUD is richer than UD. A closer look at the differences in the first double conversion shows that 82% are due to the flattening of idiomatic structures in UD, the rest coming from the ambiguity of UD in the dependencies on coordinations and nuclei.

6 A practical guide for the development of a SUD treebank

Several tools are already available for helping the start of a new treebank in SUD.

GREW-MATCH (Guillaume, 2021) is an on-line graph query tool which is dedicated to linguistic structures and in particular dependency graphs. It can be used during annotation in order to have a transversal view on already annotated data which helps to take consistent decisions on new annotations. During the maintenance of the corpora, it also helps to ensure global consistency and to do error-mining. GREW-MATCH can be easily coupled with the two UD ⇔ SUD conversion systems and gives access to the parallel view of both annotation schemes: you can search in SUD and see also the UD corresponding structure and the reverse.

The whole annotation process can be managed through the ARBORATORGREW[18] annotation platform (Guibon et al., 2020): user handling, access control, manual edition of the data... GREW-MATCH requests are also available through the ARBORATORGREW platform and detected inconsistencies can be corrected directly. In ARBORATORGREW, the user have also access to some specific tools:

- A lexicon-based view of the treebank for detecting inconsistencies in the annotation of the different occurrences of a form or a lemma
- Automatic graph transformation for the correction of regular errors or for applying changes in the annotation decisions (in the sentence-based as well as in the lexicon-based view of the treebank)

A validation page for SUD treebank is available through GREW-MATCH. It checks that structures are well-formed and helps keeping consistent decisions during the annotation process. Through the conversion to UD, the validation of the UD data adds another layer of verification. Comparing the output of the double conversion SUD ⇒ UD ⇒ SUD with the original data is an additional way to obtain valuable feedback on the annotated data.

It should be noted that in the particular case where a UD treebank already exists, the universal conversion should be tested to verify that the internal structure of the nuclei matches the expected structure. If this is not the case, the conversion may need to be customized as explained in Section 4.

7 Conclusion

SUD is not just a richer and easier annotation scheme than UD that can automatically be converted to UD. Importantly, SUD's distributional criteria facilitate and homogenize the annotation choices, resulting in treebanks that enable typological measures across languages. Also, a rich set of tools is available that allow for a kick-start in annotation of raw or partially annotated data. Several SUD treebanks exist that can serve as examples, with more in the pipeline. Go SUD!

[18] https://arborator.github.io

References

Olivier Bondéelle and Sylvain Kahane. 2021. Les particules verbales du wolof et leur combinatoire syntaxique et topologique. *Bulletin de la Société de Linguistique de Paris*, 115(1):391–465, January.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, 07.

Matthew S. Dryer. 1992. The greenbergian word order correlations. *Language*, 68(1):81–138.

Kim Gerdes and Sylvain Kahane. 2013. Defining dependencies (and constituents). *Frontiers in Artificial Intelligence and Applications*, 258:1–25.

Kim Gerdes and Sylvain Kahane. 2016. Dependency annotation choices: Assessing theoretical and practical issues of universal dependencies. In *LAW X (2016) The 10th Linguistic Annotation Workshop: 131.*

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2018. SUD or surface-syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 66–74, Brussels, Belgium, November. Association for Computational Linguistics.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2019. Improving surface-syntactic Universal Dependencies (SUD): MWEs and deep syntactic features. In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 126–132, Paris, France, August. Association for Computational Linguistics.

Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. 2020. When Collaborative Treebank Curation Meets Graph Grammars. In *LREC 2020 - 12th Language Resources and Evaluation Conference*, Marseille, France, May.

Bruno Guillaume. 2021. Graph Matching and Graph Rewriting: GREW tools for corpus exploration, maintenance and conversion. In *EACL 2021 - 16th conference of the European Chapter of the Association for Computational Linguistics*, Kiev/Online, Ukraine, April.

Hiroshi Kanayama, Na-Rae Han, Masayuki Asahara, Jena D. Hwang, Yusuke Miyao, Jinho D. Choi, and Yuji Matsumoto. 2018. Coordinate structures in Universal Dependencies for head-final languages. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 75–84, Brussels, Belgium, November. Association for Computational Linguistics.

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

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Stéphane Robert. 1991. *Approche énonciative du système verbal: le cas du wolof*. CNRS Editions.

Peter Timothy Saunders. 1980. *An introduction to catastrophe theory*. Cambridge University Press.

Karel Van den Eynde and Piet Mertens. 2003. La valence: l'approche pronominale et son application au lexique verbal. *Journal of French language studies*, 13(1):63–104.

Daniel Zeman. 2008. Reusable tagset conversion using tagset drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA).