SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, Guy Perrier

SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD

Kim Gerdes*, Bruno Guillaume†, Sylvain Kahane◊, Guy Perrier†
*LPP, Sorbonne Nouvelle & CNRS
†Loria, Université de Lorraine & CNRS & INRIA, Nancy;
◊Modyco, Université Paris Nanterre & CNRS
kim@gerdes.fr, bruno.guillaume@inria.fr,
sylvain@kahane.fr, guy.perrier@loria.fr

Abstract

This article proposes a surface-syntactic annotation scheme called SUD that is near-isomorphic to the Universal Dependencies (UD) annotation scheme while following distributional criteria for defining the dependency tree structure and the naming of the syntactic functions. Rule-based graph transformation grammars allow for a bi-directional transformation of UD into SUD. The back-and-forth transformation can serve as an error-mining tool to assure the intra-language and inter-language coherence of the UD treebanks.

1 Introduction

Universal Dependencies (UD) is an astonishing collaborative project of dozens of research groups around the world, developing an annotation scheme that is applicable to all languages and proposing treebanks based on that scheme for more than 70 languages from different language families (Nivre et al. 2016). From the start, considerable efforts have been made to avoid an anglocentric scheme, going as far as analyzing English prepositions as case markers. The project is based on an ongoing and constantly evolving collaborative construction of the annotation scheme itself by means of an open online discussion group. The project welcomes and collaborates with enrichment efforts such as the enhanced UD annotation of deep syntax (Schuster & Manning 2016) or the annotation of multi-word expressions (Savary et al. 2015).

Just as any annotation project, UD had to make choices among the different annotation options that commonly reflect opposing goals and downstream applications of the resulting treebanks. UD decided to stick to simple tree structures (compared to graphs with multiple governors) and to favor content words as heads, which is supposed to maximize "parallelism between languages because content words vary less than function words between languages" (UD Syntax: General Principles page http://universaldependencies.org/u/overview/syntax.html). The goal of "maximizing parallelism between languages" might be of use for parser development of neighboring languages, but reducing language differences makes the resulting treebank by definition less interesting for typological research on syntax. In particular, UD does not account for the hierarchy between functional words and tends to flatten syntactic structures. The content-word-centric annotation is also problematic for the internal cohesion of the treebank (cf. the difficulty of coherently annotating complex prepositions that usually contain a content word, Gerdes & Kahane 2016) and it marks a break with syntactic traditions, where headedness is defined by distributional properties of individual words (Bloomfield 1933), see Section 2.[1]

One of the central advantages of dependency grammar is the clear distinction of category (the POS, i.e. an intrinsic distributional class) and function (i.e. the specific role a word plays towards another word). Sentences such as *She became an architect and proud of it* which have given rise to a considerable amount of scholarly discussions (Sag 2003) because an X-bar based phrase structure analysis requires deciding on the category of the coordinated argument first. UD inherited from the Stanford parser[2] a mixed annotation scheme where relation labels include

[1] UD defines headedness indirectly via the category of the word: Content words are heads in UD and content words are usually understood as words belonging to open distributional classes, such as nouns, verbs, adjectives, and adverbs.

[2] The first versions of the Stanford parser were phrase structure based, providing trees that did not include functional information. The dependency output was a conversion from the phrase structure tree where the relations were computed from the category of the constituents (de Marneffe et al. 2006).

categories, as for example *nsubj* where the "n" indicates the category of the dependent. As a consequence of including the POS of the dependent in the relation name, UD has different labels for the same paradigm occupying the same syntactic position. For instance the complement of *consider* can be nominal or clausal as in *I consider this point / to leave / that you leave* and receives three different UD relation labels (*obj*/*xcomp*/*ccomp*).

We propose a new surface-syntactic annotation scheme, similar to UD, that we name SUD for *Surface-syntactic Universal Dependencies*. We want dependency links as well as the dependency labels to be defined based on purely syntactic criteria (Mel'čuk 1988), giving dependency structures closer to traditional dependency syntax (Meaning-Text Theory, Mel'čuk 1988; Word Grammar, Hudson 1984, 2007; Prague Dependency Treebank, Hajič et al. 2017) and headed constituency trees in phrase structure grammar (X-bar Syntax, Jackendoff 1977; Penn Treebank, Marcus et al. 1993). We also propose a hierarchy of SUD dependency relations that allows for under-specifications of dependency labeling.

We conceived the SUD scheme as an alternative to UD and not as a competing annotation scheme, which means that the annotation scheme should have the same information content, the information being only expressed another way. Put differently, we looked for an annotation scheme based on distributionial criteria with an elementary conversion going both ways without loss, i.e. an "isomorphic" annotation. Since the principles underlying SUD are different, the isomorphism with UD cannot be perfect. As a result, SUD is near-isomorphic to UD, and we have developed two treebank conversion grammars for the Grew platform (http://grew.fr, Bonfante et al. 2018): UD to SUD and SUD to UD. We will evaluate the differences between a UD treebank and the results of a double-conversion through SUD in Section 4.

SUD treebanks can be obtained by simple conversion from UD treebanks and can be useful for teaching and typological studies. Inversely, annotations can be done directly in SUD, and ultimately converted into UD. SUD annotations are less redundant and more economical than UD annotations. For instance SUD uses a simple *subj* relation because the nominal character of a subject should be indicated only once (as a POS). The distinction between clausal and nominal subjects can be recovered automatically from the POS of the subject and its context, but how this context is taken into account depends on the language.[3]

The conversion tool Grew and the conversion grammars are freely distributed, and we envision to propose the UD treebanks also under the automatically converted SUD scheme on the UD website.[4] This SUD annotation scheme proposal could benefit from future discussions and evolutions of the UD ecosystem.

As a side effect, the double UD→SUD→UD conversion provides a powerful error mining tool for UD treebanks. Trees that are not stable under this conversion very often contain non-standard uses of the UD annotation scheme that deserve special attention.

Section 2 explain what is surface syntax, what are the criteria defining a surface syntactic structure and how such a structure differs from UD trees. Our SUD annotation scheme is introduced in Section 3. The conversion between UD and SUD is presented in Section 4 and evaluated on the whole set of UD treebanks.

2 Surface Syntax

We will present defining criteria for a surface syntactic analysis following Mel'čuk 1988 who proposes three types of criteria: A: When to connect two words? B: Who is the governor in a connection? C: How to classify the dependencies?

2.1 Criteria for structural choices

The basic type A criterion is the stand-alone property or "autonomizability": Two words are connected if they can form a speech turn. For example in the sentence *The little boy talked to Mary* "the boy" or "to Mary" can stand alone with the same meaning, for instance as an answer to a question such as *Who talked to Mary?* or *Who did the little boy talk to?*. Autonomizability is not sufficient to determine a dependency structure as the set of connections does not necessarily form a tree, and we need further structural criteria to decide which links to preserve (Gerdes & Kahane 2011).

For instance, there are no simple criteria to establish a connection between *talk* and *to* or *talk* and *Mary* because both *talk to*, and *talk Mary* are ungrammatical speech turns. This connection can

[3] The clausal character of a phrase is more or less explicit depending on the language. If a language allows for clauses without subjects, without subordinating conjunctions, or without verbs, the conversion SUD → UD has to be adapted accordingly. If all three indicators are absent while the clause-noun distinction remains relevant, we would have to rely on an additional feature in SUD in order to obtain a correct transformation.

[4] For the time being, the SUD treebanks are available on https://gitlab.inria.fr/grew/SUD

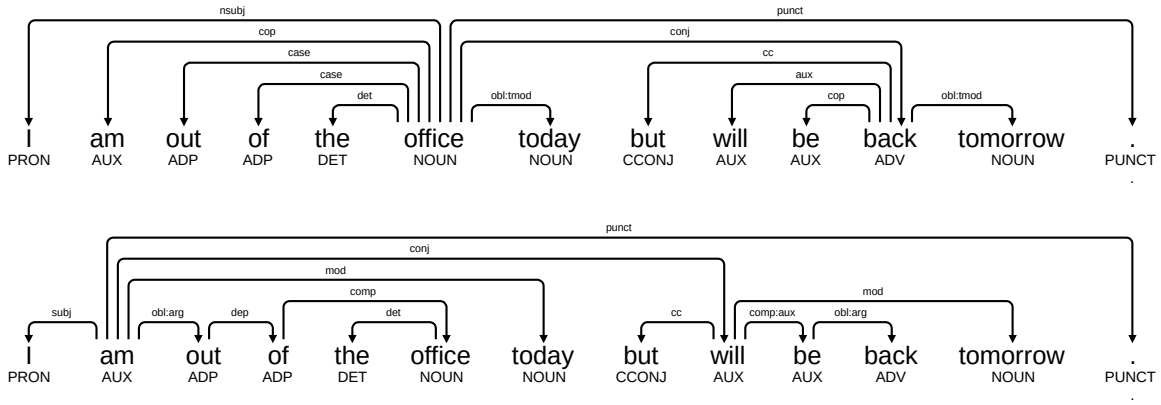

Figure 1: UD and SUD analysis of the same sentence (UD_English-EWT@2.2 email-enronsent38_01-0114)

be established by means of criteria of type B determining who, *to* or *Mary*, is the head of *to Mary*. At this point, UD parts with surface syntax criteria and applies the criterion of "content word as head" whereas surface syntax uses distributional criteria of each individual word. The main criterion is that **the surface syntactic head determines the distribution of the unit**. For instance, *Mary* and *to Mary* have a profoundly different distribution as they can never commute:

Mary slept vs. **To Mary slept*.

The boy talked to Mary vs. **The boy talked Mary.*

This suffices to show that *Mary* is not the head. Although we cannot test whether *to* has the same distribution as *to Mary* because a preposition such as *to* never appears without a noun or a verb, we consider *to* to be the head, a choice that is consistent with most if not all theoretical syntactic frameworks.[5] The same reasoning can be applied to the auxiliary-verb connection such as *has chosen* or the copula-adjective connection such as *is happy*: *chosen* never commutes with *has chosen*.[6]

A less clear case of function words as heads is the case of a conjunct in a coordination: *I invited Bill and Mary*. In most positions, *Mary* and *and Mary* cannot commute (again *and* cannot stand alone and cannot be tested). Here a second distributional criterion can be used: A dependent does not change the distribution of its governor. This shows that *Mary* cannot be considered as a dependent of *and*, because the commutation of *Mary* with units of other POSs (*and red*, *and is sleeping*, etc.) completely changes the distribution of the phrase.

Note that the case of the determiner-noun connections is less clear-cut. Both UD and traditional surface syntax (Mel'čuk 1988) chooses the noun as the head although *boy* and *the boy* do not have the same distribution. The DP analysis makes the opposite choice (Hudson 1984, 2007, Abney 1987). For these two controversial cases, we keep the UD analysis with the functional word as a dependent.

As an illustration of the flat UD structures compared to SUD, consider Figure 1 showing the analyses of *I am out of the office today but will be back tomorrow.* The UD tree has depth 3 and a maximum number of 8 dependents per node whereas the SUD tree has depth 5 and only a maximum number of 5 dependents per node. We generalize this observation into a general principle: We believe that the syntactic structure follows the *dependency length minimization principle:* "Languages tend to minimize the surface syntactic dependency length" because this reduces the cognitive load of language processing (Liu 2008, Futrell et al. 2015). We use this argument to attach each conjunct to its closest neighbor conjuncts and to attach shared dependents to the closest conjunct. This gives us a chaining analysis of coordination instead of UD's bouquet analysis.[7] Figure 2 shows an example that illustrates the structural differences for coordination between UD and SUD.

[5] The tokenization is quintessential here. If an annotation scheme of a inflectional language decides to separate case markers, such a case marker will become the head of the word (Groß 2011).

[6] If the dependent of an *aux* relation is optional, invariable, and non-verbal, it should be tagged as PART. Then it will not be promoted to the head-position in the UD → SUD conversion.

[7] One of the arguments in favor of a bouquet analysis is to allow the disambiguation of embedded coordinations such as *A and B or C*: For (*A and B*) *or C*, *or C* depends on *A*, while for *A and* (*B or C*), *or C* depends on *B*. Nevertheless, this disambiguation is partial because in case of a flat coordination such as *A, B, or C*, we see that or *C* also depends on *A* and thus, the bouquet structure cannot distinguish the embedded (*A and B*) *or C* situation from the flat *A, B, or C* situation.

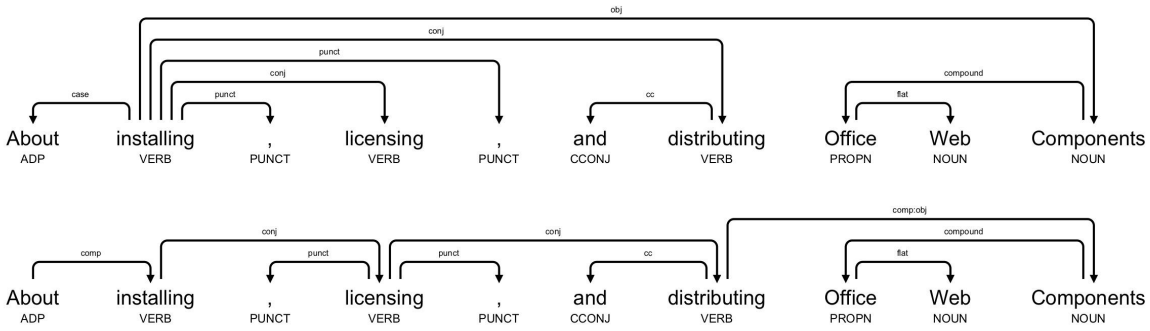

Figure 2: Coordination in UD and in SUD
(UD_English-LinES@2.2 257, comma attachment as in the original treebank).

2.2 Criteria for dependency labels

We need criteria to decide whether two dependencies (in two different sentences) must be labeled by the same relation or not. Our first criterion allows us to decide when the governors are the same: If two dependents occupy the same position, i.e. they are mutually exclusive, they must be connected to their governor by the same relation.[8] This implies that in *This apple is good for you* and *Eating this apple is good for you*, both *this apple* and *eating this apple* must have the same function. Equally, *that apple* and *to eat that apple* have the same function in *I want that apple* and *I want to eat that apple.* This criterion is currently not used in UD (cf. *nsubj* vs. *csubj* for subjects and *obj* vs. *xcomp* for objects).

Our second criterion is used to decide whether a dependent D that appears in two governor-dependent relations in two different sentences should be labeled with the same function: The relations are the same if the paradigms of units that can commute with D in the two positions are roughly the same, semantic constraints apart. As an example of a semantic selectional restriction, we establish the same *subject* positions for "think" and "sink" although the paradigms are not exactly the same: *the boat sinks* vs. ???*the boat thinks*.[9] Inversely, the French verbs *parler* 'talk' and *penser* 'think' both have a complement with the preposition *à* 'to', but the pronominalization of these arguments is different: *parler à Mary* 'talk to Mary' → *lui parler* 'speak to her' vs. *penser à Mary* 'think of Mary' → *penser à elle* 'think of her'. This could lead us to distinguishing two types of arguments (e.g. "indirect object" vs. "oblique complement").[10]

Two positions only rarely have exactly the same paradigms and constraints, but they can be more or less similar. Thus, the notion of function is not absolute but rather relative, which justifies a hierarchy of functions, thus allowing for choosing between coarser or finer grained analyses.

Although, as we have shown, UD has a tendency to use several relation labels for the same syntactic function, the UD annotation scheme can also combine two syntactic functions into one: For example, all PP dependents of a verb are connected with the same relation *obl* to their governor, conflating prepositional arguments and repeatable modifiers.[11]

3 SUD

With this basis, we have developed an annotation scheme that attempts to remain as close as possible to the UD annotation scheme while obeying to surface-syntactic criteria. The SUD annotation scheme is a surface-syntax annotation

[8] The inverse is not a necessary condition: We can decide to group together under one relation label two dependents that can co-occur with the same governor, in particular modifiers of verbs or of nouns, which can be repeated.

[9] Put differently, the set of elements that can occupy *sink*'s subject position and the set of elements that can occupy *think*'s subject position are different. But the two sets are sufficiently similar and the restriction seems to be of semantic nature that we decide not to introduce an "animate-subject" relation and an "inanimate-subject" relation, but to simply use the *subj* function for these verbs' first positions.

[10] A third criterion states that redistribution and agreement constraints for both dependency should be the same. As an example of different redistributions, consider *cost* vs. *win*: *Peter won 100€* can be passivized but not *The book costs 100€*. Accordingly, an annotation scheme can decide to establish two distinct functions (e.g. "direct object" vs "measure complement").
In SUD, we unite all these cases under the function name *comp*, see Section 3.1, therefore not distinguishing "indirect objects" from "oblique complements" or "direct objects" from "measure complements".

[11] Several UD treebanks decided to keep the verbal valency and thus to mark the distinction between prepositional arguments and modifiers, for example by means of *obl:arg* vs. *obl:mod*, such as Arabic, Cantonese, Chinese, Czech, French, Hindi, Polish, Sanskrit, and Slovak. The secondary annotation label of this argument vs. modifier distinction has not yet been unified across languages and some treebanks use *:tmod*, *:npmod*, and *:loc* vs. *:agent* among others.

scheme, which implies in particular that: 1. Contrarily to UD, function words such as adpositions, subordinating conjunctions, auxiliaries, and copulas are heads. 2. Words that are in the same paradigm of commutation (and thus occupy the same syntactic position) have the same function, i.e. they are connected to their governor by the same syntactic relation.

3.1 Structural choices

In a nutshell, UD's *case*, *mark*, *aux*, and *cop* dependencies are inverted while other UD dependency directions are preserved. In particular, we kept coordinating conjunctions and determiners as dependents (see Section 2.1).

The directional changes of a relation opens the question of the attachment of the dependents involved in the changes. In UD, function words do not have dependents, but in surface syntax, modifiers of the whole phrase are traditionally attached to the head, which can now be a function word. Put differently, we have to decide which dependents are attached to the function word and which remains on the lexical word. It is generally accepted that the subject is a dependent of the auxiliary or the copula, with whom it agrees in inflectional languages. Highly grammaticalized elements such as negation should go onto the auxiliary whereas arguments should remain on the lexical element. For the sake of simplicity, all modifiers have been attached on the auxiliary in SUD and all arguments except the subject remain on the lexical verb.[12] Conjuncts need special rules to be handled correctly, because sometimes they must be raised (*Mary was sleeping and knew it*) and sometimes not (*Mary was sleeping and snoring*).

3.2 Labeling choices

SUD introduces four new relations: *subj*, *comp*, *mod*, and *unknown* and reassign a more specific meaning to the *dep* label. All subjects have the function ***subj***, grouping together UD's *nsubj* and *csubj*. All other arguments of adjectives and verbs have the function ***comp***, bundling UD's *obj*, *iobj*, *xcomp*, and *ccomp; comp* is also used for all complements of function words such as auxiliaries, copulas, adpositions, and subordinating conjunctions, thus replacing UD's *aux*, *cop*, *case*, and *mark*. Modifiers have the function ***mod*** wherever we can clearly distinguish the modifiers from arguments. If not, we use the ***dep*** relation to indicate that we cannot.[13] This ***dep*** relation is particularly useful for PP attachments to nouns but also for UD's *obl* relation if it is not specified further as *obl:arg* or *obl:mod*. If we have the argument-modifier distinction for PP dependents of verbs we classify *obl:arg* as *comp* and *obl:mod* as *mod*. If the nature of the relation cannot be determined, we use the ***unknown*** label (where UD used the *dep* label), which becomes the hypernym of all SUD relations (Figure 3).

Compared to UD we thus grouped together relation labels whenever the distinction between them is purely categorical, i.e. contingent on the POS of the governor or the dependent. To avoid annotation redundancy, we do not use UD's *acl*, *advcl*, *advmod*, *amod*, *aux*, *case*, *ccomp*, *cop*, *csubj*, *iobj*, *mark*, *nmod*, *nsubj*, *nummod*, *obj*, *obl*, and *xcomp* relations. All other UD relation labels are preserved.

SUD dependency	Corresponding UD dependencies
dep	*acl, amod, nmod, nummod, obl*
comp	*aux, ccomp, iobj, obj, obl:arg, xcomp, cop, mark, case*
mod	*advcl, advmod, obl:mod*
subj	*csubj, nsubj*

Table 1: SUD and corresponding UD relation labels

As a general principle of allowing a varying granularity of dependency relation labels, but also to assure the convertibility with UD, SUD relies heavily on secondary relation labels that are, just like in UD, separated from the main label by a colon: *primary:secondary*. These secondary labels are optional in a simple native SUD annotation but necessary for a full convertibility into UD. On the contrary, the converted SUD uses the distinction between *comp:aux* and *comp:pass* to discriminate the complement of an AUX used as a tense auxiliary and as a passive auxiliary, and it also uses *comp:cop* or *comp:caus* for the conversion of UD's *cop* and *aux:caus*. The UD relations *iobj* and *obl:arg* both give *comp:obl* in SUD, *ccomp* and *obj* give *comp:obj*, and *xcomp* gives *comp:rais* (Table 2).[14]

[12] A native SUD annotation might choose to propose more specific rules defining the distribution of modifiers between the function verb and the lexical verb. This has no incidence on the automatically obtained corresponding UD analysis, because such a distinction is flattened when converting into UD.

[13] The *dep* relation thus becomes a hypernym of *comp*, *mod* and *subj*, as well as *cc* and *det*.

[14] Although *comp:obj* and *comp:obl* are clearly subfunctions of *comp*, this is not stricto sensu the case of *comp:rais*. For example, we consider that (Fr.) *dormir* 'to sleep' and *que tu dormes* 'that you sleep' have the same function *comp:obj* in the context *Je veux* _ 'I want _', while *que tu dormes* has a different function *comp:obl* in the context *Je m'étonne* _ 'I'm surprised _', where it commutes with a PP *de ça* 'of that'. A native SUD annotation could thus distinguish *comp:obj:rais* from *comp:obl:rais* by means of triple labels.

UD dependency	SUD dependency	UPOS of the governor	UPOS of the dependent	Other relations starting on the dependent
obl	dep	ADJ|VERB		
acl		NOUN|PROPN|PRON	ADP	comp->VERB
		NOUN|PROPN|PRON	VERB	
amod			ADJ	
nmod		NOUN|PROPN|PRON	ADP	comp->NOUN|PROPN|PRON
nummod			NUM	
advcl	mod	ADJ|VERB	ADP	comp->VERB
			ADJ|VERB	
advmod			ADV	
obl:mod			ADP	comp->NOUN|PROPN|PRON
obj	comp:obj		NOUN|PROPN|PRON	
ccomp			VERB	
			SCONJ	comp->VERB
	comp:obl		SCONJ	comp->VERB
iobj			PRON	
obl:arg			ADP	comp->NOUN|PROPN|PRON
			ADV	
csubj	subj		VERB	
nsubj			NOUN|PROPN|PRON	
xcomp	comp:rais			

Table 2: UD-SUD transformation correspondences

4 Convertibility between UD and SUD

The conversion UD → SUD is done in three main steps: 1) transforming the bouquet structure into a chaining analysis (for relations *conj*, *fixed* and *flat*); 2) reversing relations *aux*, *cop*, *mark* and *case*; 3) mapping UD relations to SUD relations following Table 2. The reverse conversion (SUD → UD) also proceeds in three steps in the same vein.

The second step is the most problematic because a lexical head can have several function words depending on it (up to 7 in UD_Japanese!). In such a case, we must decide which one depends on which one.

To do this, we rely on a universal hierarchy of relations that the auxiliaries have with the main verb, in particular *mark* relations are higher than *aux* relations and time and aspect auxiliaries are higher than voice auxiliaries (Van Valin 1984, Cinque 1999). When this information is unavailable we rely on the word order: The closest function word is the SUD governor of the lexical head, the next one is the SUD governor of the first one, and so on.

The conversions (UD → SUD and SUD → UD) we proposed are encoded in a rule-based system. The rules are organized by means of a separation of a universal core rule set and a language specific rule set, which for the time being has only been implemented for French.

We use the Grew software (http://grew.fr) based on a computational Graph Rewriting Model. Each conversion is encoded as a graph rewriting system (GRS): a set of rules and a strategy describing how the rule applications must be ordered. Below, we give an example of an UD → SUD rule for the inversion of *mark*:

```
rule left_mark {
  pattern { e:H-[mark]->X1; X1 << H }
  without { H-[aux|aux:pass|aux:caus|cop|
    mark|case]->X2; X1 << X2 }
  commands {
      del_edge e;
      add_edge X1-[comp]-> H;
      shift_out H =[aux|aux:pass|aux:caus|
        cop|mark|case|conj|cc|root]=> X1; } }
```

The rule contains three parts: the *pattern* part says that the rule applies on a dependency *e* labeled *mark*, with a dependent X1 preceding its head H; the *without* part ensures that there is no other element *aux*, *cop*, *case* or *mark* depending on H between X1 and H; the *commands* part describes the required modifications on the structure: delete the matched edge *e*, add a new edge *comp* in the reverse order, and the *shift_out* command gives the list of relations that must be moved from node H to node X1. It is worth noting that *aux*, *case*, *cop*, and *mark* that remain to be inverted must be raised onto the auxiliary.

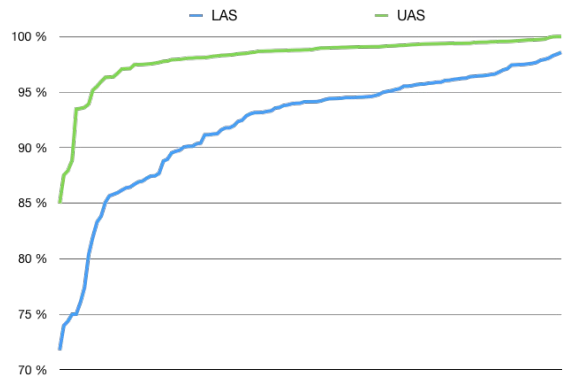

Figure 3: LAS and UAS of UD→SUD→UD transformations across the UD 2.2 treebanks, displayed on the X-axis by ascending LAS (resp. UAS) order.

We have evaluated the results of the double conversion (from UD to SUD first and then from SUD back to UD) against the original UD annotation with the 122 corpora of version 2.2. The experiment were conducted on the test part of each corpus. The median value of the LAS scores is 94.1%. Three corpora have a LAS score below 75%: UD_Korean-Kaist (71.8%), UD_Japanese-BCCWJ (74.0%) and UD_Japanese-GSD (74.4%). The 3 highest values are for UD_Hungarian-Szeged (98.6%), UD_Italian-ParTUT (98.4%), and UD_Italian-PoSTWITA (98.3%). The median value of the UAS scores is 98.8%. The 3 lowest scores are for UD_Yoruba-YTB (85.0%), UD_Japanese-GSD (87.5%) and UD_Japanese-PUD (87.9%). Two corpora have a 100% UAS score: UD_Warlpiri-UFAL and UD_Telugu-MTG.

Figure 4 shows the distribution of LAS (blue curve) and UAS (green curve) on the 122 treebanks. The two curves present the ordered set of values of LAS/UAS (not corresponding to the same corpus ordering). Although the scores are very high, the procedure does not allow to evaluate the two conversion systems separately: A dependency may remain unaffected by both conversions when it should have been, and this error will not be detected.

One central source of the discrepancy between a corpus and its double conversion is the inconsistency between a relation name and the POS of its dependent. For instance, the conversion UD→SUD always produces *dep* for an *amod*, but the SUD→UD is not able to recover *amod* if the dependent is not an ADJ. In the corpus with the lowest LAS score (UD_Korean-Kaist), we observed many unusual associations of relation and POS. In the whole corpus UD_Korean-PUD, 22.4% of the *advmod* relations have a dependent that is not an ADV, and 43.5% of the *aux* relations have a dependent which is not an AUX. In the corpus UD_Korean-PUD, all the 323 *aux* relations have a dependent which is not an AUX. Until now, we have only designed a set of generic rules that may be refined for each language and it is difficult to draw conclusions about the full set of corpora.

A part of these inconsistencies may also be linked to MWEs: An MWE as a whole often has a POS which is different from the POS of its first token. In UD 2.2, 4 corpora contain the feature MWEPOS to annotate the POS of the MWEs (the conversion in the evaluation curves above does not uses this feature). If this information is taken into account in the conversions, the LAS scores significantly increase in 3 of the 4 cases (UD_French-Sequoia: +1.05%, UD_Catalan-AnCora: +0.80%, UD_Spanish-AnCora: +0.75% and UD_Portuguese-Bosque: +0.08%).

We believe that a further exploration of these inconsistencies could provide a crucial step for the improvement of the treebanks as well as the conversion rules. As a next experiment, we plan to introduce a new feature UDPOS to add the expected POS where the current UD POS is unexpected. Then, each UDPOS have to be interpreted as: 1) an annotation error, 2) a place where a MWEPOS is missing, or 3) a special usage of the relation that should be taken into account in the language specific conversion rules.

5. Conclusion

Based on UD, we propose a new annotation scheme, SUD, which follows standard distributional criteria for headedness and relation labeling and is thus closer to traditional constituency-based surface syntax as well as to dependency-based surface syntax. This means in particular that this new scheme can be employed more easily by users and annotators that are trained in more traditional forms of syntax. As an experiment, we are now developing a new treebank directly in SUD and this treebank will subsequently be converted into UD, the automatic transformation providing a quality and coherence control of the SUD annotation.

Such a format is useful for every computation that concerns the form of the sentence such as word order (Chen et al. submitted) and the relation to prosody, etc. Conversely, UD might be a better entry point to the semantic content of the sentence.

The lower dependency length gives psycholinguistic support to SUD treebanks. Possibly related is the fact that various experiments on parser performance also

consistently give an advantage to function-word-headed structures (Schwartz et al. 2012, Silveira and Manning 2015, Kirilin and Versley 2015, Rehbein et al. 2017)[15] which provides another *raison d'être* for parallel SUD treebanks.

The whole UD 2.2 database, with its 122 treebanks, has been converted into SUD and is already accessible at https://gitlab.inria.fr/grew/SUD. We would like to see this alternative to be distributed on the UD website as soon as possible and hope that the new scheme will benefit from discussions with the whole community and evolve in parallel to the UD scheme. Then SUD would become an alternative annotation option for UD treebank developers.

As a last point, it appears that the conversion between UD and SUD sheds light on some potential problems in UD treebanks. We have to better understand why the double conversion UD→SUD→UD gives bad results on some treebanks and to what extent this is due to problems in our conversion grammar, or rather caused by an unexpected usage of the UD scheme that could be fixed, either by correcting the treebank or by adapting the annotation reference guide to include and standardize the new analyses of a given construction. It might be useful to adapt the SUD conversion for each language, which could eventually allow for isomorphic transformations.[16] Making the UD treebanks SUD compliant would lead to a more homogeneous annotation and could lead the way in the ongoing discussion towards the upcoming UD 3.0 annotation scheme.

References

Leonard Bloomfield. 1933. *Language*.

Xinying Chen, Kim Gerdes, Sylvain Kahane. Submitted. Typometrics: From Implicational to Quantitative Universals in Word Order Typology.

Guglielmo Cinque, 1999. Adverbs and functional heads: A cross-linguistic perspective. Oxford University Press.

Marie-Catherine de Marneffe, Bill MacCartney, Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. *Proceedings of LREC*.

Richard Futrell, Kyle Mahowald, 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, Sylvain Kahane. 2011. Defining dependencies (and constituents). *Proceedings of the First International Conference on Dependency Linguistics (Depling 2011).*

Kim Gerdes, Sylvain Kahane. 2016. Dependency annotation choices: Assessing theoretical and practical issues of universal dependencies. *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016).*

Thomas Groß. 2011. Catenae in morphology. *Proceedings of the First International Conference on Dependency Linguistics (Depling 2011).*

Guillaume Bonfante, Bruno Guillaume, Guy Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*. John Wiley & Sons.

Jan Hajič, Eva Hajičová., Marie Mikulová, Jiří Mírovský. 2017. Prague Dependency Treebank. *Handbook of Linguistic Annotation*. Springer, Dordrecht. 555-594.

Ray Jackendoff. 1977. *X-bar syntax: A Study of Phrase Structure*, Linguistic Inquiry Monograph 2. Cambridge, MA: MIT Press.

Angelika Kirilin, Yannick Versley. 2015. What is hard in Universal Dependency Parsing. *Proceedings of the 6th Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2015)*.

Haitao Liu. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science, 9(2)*, 159-191.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics 19.2:* 313-330.

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

Ines Rehbein, Julius Steen, Bich-Ngoc Do. 2017. Universal Dependencies are hard to parse–or are they? *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017).*

Ivan A. Sag. 2003. Coordination and underspecification. *Proceedings of the 9th International Conference on HPSG*.

Agata Savary, et al. 2015. PARSEME–PARSing and Multiword Expressions within a European multilingual network. *7th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC 2015).*

Sebastian Schuster, Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An

[15] Although the advances in neural parsers trained on sufficient data probably will ultimately wipe out any differences (because the parser can independently learn the UD-SUD transformation rules).

[16] This aspect distinguishes SUD from Enhanced UD. The latter is clearly a new level of annotation with additional information. On the contrary, SUD is an alternative encoding of UD information.

Improved Representation for Natural Language Understanding Tasks." *Proceedings of LREC*.

Roy Schwartz, Omri Abend, Ari Rappoport. 2012. Learnability-based syntactic annotation design. In *Proceedings of COLING 24*, 2405–2422.

Natalia Silveira, Christopher Manning. 2015. Does Universal Dependencies need a parsing representation? An investigation of English. *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*.

Robert D. Van Valin Jr. 1984. A typology of syntactic relations in clause linkage. In *Annual meeting of the Berkeley Linguistics Society* (Vol. 10, pp. 542-558).