All Roads Lead to UD: Converting Stanford and Penn Parses to English Universal Dependencies with Multilayer Annotations

Siyao Peng
Department of Linguistics
Georgetown University
sp1184@georgetown.edu

Amir Zeldes
Department of Linguistics
Georgetown University
amir.zeldes@georgetown.edu

Abstract

We describe and evaluate different approaches to the conversion of gold standard corpus data from Stanford Typed Dependencies (SD) and Penn-style constituent trees to the latest English Universal Dependencies representation (UD 2.2). Our results indicate that pure SD to UD conversion is highly accurate across multiple genres, resulting in around 1.5% errors, but can be improved further to fewer than 0.5% errors given access to annotations beyond the pure syntax tree, such as entity types and coreference resolution, which are necessary for correct generation of several UD relations. We show that constituent-based conversion using CoreNLP (with automatic NER) performs substantially worse in all genres, including when using gold constituent trees, primarily due to underspecification of phrasal grammatical functions.

1 Introduction

In the past two years, the Universal Dependencies project (UD, Nivre et al. 2017), offering freely available dependency treebanks with a unified annotation scheme in over 50 languages, has grown rapidly, allowing for cross-linguistic comparison and computational linguistics applications. At the same time, because of its rapid growth and the need to negotiate annotation schemes across languages, annotating large resources from scratch in the latest UD standard is challenging, not only because of the annotation effort, but also because guidelines may change mid-way, and data and annotator training must be revisited to match the latest developments. Instead, a large number of projects within UD capitalize on existing treebanks converted from constituent treebanks (in English usually using CoreNLP, Manning et al. 2014) or other dependency schemes, meaning that for those projects that are not annotated directly in UD, changes to the UD guidelines generally mean adapting an existing converter framework.

In this paper, we concentrate on English dependency treebanking, which has been dominated by data converted from Penn Treebank-style constituent trees (cf. Bies et al. 1995). We compare results of constituent treebank conversions with results from converting English dependency data annotated using the older (and by now frozen) Stanford Typed Depenendencies (hence SD, de Marneffe and Manning 2013). Specifically, we will be working with the freely available Georgetown University Multilayer corpus (GUM, http://corpling.uis.georgetown.edu/gum/), which we have converted to the latest UD standard (as of UD version 2.2). The paper has several goals:

1. To describe and evaluate the accuracy of gold standard SD to UD conversion (SD2UD)
2. To explore the necessary layers of annotation for generating gold UD from gold SD data, including information that is not strictly present in the syntactic parse
3. Comparing conversions from SD source data and constituent tree source data
4. Making a substantial new English resource, with over 85,000 tokens in 8 genres, available in UD

We will show that while rule-based SD to UD conversion is already highly accurate, it must also rely on multiple annotation layers outside of the parse proper if the full range of dependencies is targeted. For the third goal in particular, our evaluation of the converted UD product reveals that 'native' dependency

data in English differs from converted constituents in several ways, including the presence of some rare labels and the proportion of non-projective dependencies.

2 Corpora

The main corpus used in this paper is the Georgetown University Multilayer corpus (GUM, Zeldes 2017), a freely available corpus covering data from eight English genres: news, interviews, how-to guides, travel guides, academic writing, biographies, fiction and web forum discussions. The corpus is annotated by students at Georgetown University[1] and currently contains 101 documents, with over 85,000 tokens, annotated for:

- Multiple POS tags (Penn tags, Santorini 1990, TreeTagger tags and CLAWS5 tags, Garside and Smith 1997), as well as lemmatization
- Sentence segmentation and rough speech act (based on SPAAC, Leech et al. 2003)
- Document structure (paragraphs, headings, etc.), ISO date/time annotations and speaker information
- Gold SD dependencies and automatic constituent parses based on gold POS tags
- Information status (given, accessible and new, based on Dipper et al. 2007)
- Entity and coreference annotation, including bridging anaphora
- Discourse parses in Rhetorical Structure Theory (Mann and Thompson 1988)

A second English corpus we will be comparing this data to in Section 4.4 is the English Web Treebank (Bies et al. 2012, Silveira et al. 2014), containing over 1,170 documents with over 250,000 tokens in five genres: blog posts, e-mails, newsgroup discussions, online answer forums and online reviews. This corpus was originally annotated using Penn-style constituent trees and converted to UD using CoreNLP (Schuster and Manning 2016), with subsequent scripts and manual corrections producing the version now available in UD V2.2.

3 Method

In this section we focus on describing our approach to converting SD parses to UD with and without supplemental information from further layers of annotation. The evaluation in Section 4 will compare these scenarios with several conversion scenarios from constituent trees.

3.1 SD conversion rules

Our conversion process comprises three parts:

1. a preprocessing step pulling in information from annotation layers outside of the syntax tree proper
2. the main rule-based conversion
3. a postprocessing step in which punctuation is attached using the freely available udapi API (Popel et al. 2017)

This section concentrates on the main, syntactic rule-based conversion, while the next section focuses on information brought in from other annotation layers.

The main step uses a configurable rule-based converter called DepEdit[2] which allows the definition of conversion rules, each having three components: 1. a set of key-value pairs denoting regular expressions matching targeted token properties; 2. a set of relations which must hold between these tokens; and 3. instructions on how to alter token properties when the rule is matched. Some example rules are given in Table 1.

[1] For an analysis of annotation quality and genre differences within the corpus, see Zeldes and Simonson (2016)

[2] Available at `https://corpling.uis.georgetown.edu/depedit/` and via PyPI (`pip install depedit`).

attributes	relations	actions
func=/dobj/	none	#1:func=obj
func=/.*/;func=/^cc$/;func=/^conj$/	#1>#2;#1>#3	#3>#2
func=/prep/;pos=/^W.*/;func=/pcomp/	#1>#3;#3>#2	#2:func=pobj;#1>#2;#2>#3;#3:func=rcmod

Table 1: Examples of DepEdit rules

The first example illustrates a trivial renaming rule, in which the SD label *dobj* is renamed to UD *obj*: the definition in the first column matches any token with a function label matching `/dobj/`, no relations are imposed (`none`), and the action specifies that the first (and only) token in the definition, #1, should have its function label set to *obj*. Similar rules are used to create Universal POS tags, which is almost trivial, since the corpus already contains gold Penn Treebank-style POS tags and lemmas. However, in some cases, dependency relations must be consulted too, e.g. the verb 'be' must be given the AUX tag as a copula or auxiliary, and otherwise VERB; determiners (e.g. *that*) become DET when modifying nouns, but are PRON when used independently; etc.

The second example in Table 1 is more complex and changes the graph in Figure 1 from the coordinating conjunction 'and' being governed by the first conjunct (SD guidelines) to being governed by the second (UD V2.2 guidelines). The attribute definitions first specify 'any function' (`func=/.*/`), then for a second token (separated by ';') that its function must be *cc* (coordinating conjunction), followed by a third token labeled *conj*. The relations column then specifies that token #1 governs #2 and that it also governs #3. Finally the actions column specifies that #3 should now govern #2, leaving unchanged the fact that #1 governs #3. The process of applying these two rules is shown for a fragment in Figure 1, where the source (SD) graph is rendered above the tokens, and the result (UD) below, rendered in blue.

The third rule handles free relative clauses, and targets WH pronouns governed by a preposition and *pcomp*, in constructions such as "an expectation of$_{\#1}$ what$_{\#2}$ to do$_{\#3}$", which should be converted to a relative clause (in SD, *rcmod*). Note that since this rule occurs before conversion of prepositions to the UD label *case* and relatives to *acl*, SD labels are still used in this rule. POS substitutions are also cascaded, meaning rules can initially refer to Penn tags, and later on to UPOS tags.[3]

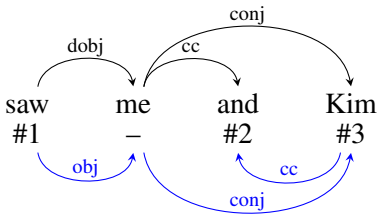

Figure 1: Converting coordination from SD to UD

The most current set of conversion rules, numbering nearly 100 items, can be found along with conversion utilities is freely available online.[4]

3.2 Using multilayer annotations

The availability of several kinds of non-syntactic gold annotations in GUM allows us to refine the conversion process further. While it could be argued that syntax trees should not contain non-syntactic information to begin with, UD parses do in fact integrate information which seems to be not completely syntactic, and more so than SD: specifically, as we will see below, factors such as ontological entity types, coreference information and presence of errors or disfluencies all affect the analysis in UD. This can be

[3]Morphological features, by contrast, are generated at the end of the process using CoreNLP, as is the case for EWT. Their accuracy is not evaluated in this paper.

[4]`https://github.com/amir-zeldes/gum/blob/dev/_build/utils/stan2uni.ini`

viewed as an advantage of UD trees once the information is available, but also as an unfair requirement from parsers and converters attempting to generate data in the UD scheme.

One of the most widespread changes not recognizable from pure SD dependencies is the conversion of SD *nn* (noun modified noun) into one of two structures: *compound* for nominal compounds with internal syntactic structure and *flat* for headless multi-word expressions that are not part of the closed list receiving the label *fixed*. In practice, the *flat* label in English usually translates to proper nouns supplying names.

The large majority of *flat* cases correspond to names of persons, while most named non-persons retain a syntactic head (usually on the right).[5] This means that knowing entity types can be crucial. For example, knowing that *World Bank* is an *organization* in Figure 2 induces the *compound* relation between the two tokens; by contrast, in Figure 3, *Frank Bank*, annotated as a *person* entity on another annotation layer results in a *flat* UD annotation.[6] The preprocessing step reads entity annotation information from parallel files and flags the (SD) head of each entity mention with its entity type, which is then used in the DepEdit conversion rules. The entity's head token is matched by finding a token in the entity span which is either the sentence root, or is governed by a non-punctuation parent from outside the span.

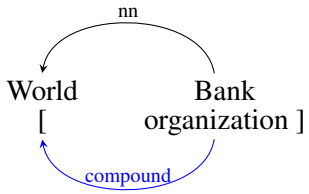

Figure 2: Converting 'World Bank' (organization)

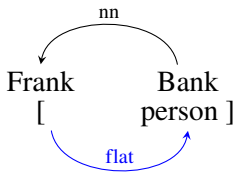

Figure 3: Converting 'Frank Bank' (person)

However, the mapping between dependencies and entity types is not one-to-one, meaning some errors are inevitable even with gold entity information. For example, some company names arguably do not exhibit internal syntactic structure and should be annotated as *flat* in UD, for example *Wells Fargo* in Figure 4. Currently, our automatic conversion will erroneously label such cases as *compound* (see Section 4.3 for error analysis).

A second type of information is required by the introduction of the label *dislocated* in UD. Although dislocation, shown in (1), is ostensibly a syntactic operation, it appears very much like any kind of topicalization, as shown in (2). Both types are annotated as *dep* in the GUM SD annotations, for lack of a better label.

(1) We like pets. [**My neighbors**$_{dislocated}$], their pets drive [**them**$_{obj}$] nuts

(2) We like [**canned foods**]. My neighbors$_{dep}$, their pets eat [**them**$_{obj}$] every day

[5]The other main category containing *flat* names is place names, but the majority of multi-word place names are nevertheless headed, and therefore labeled *compound*. A discussion on whether or not proper names such as 'Kim King' should be treated as non-headed, or arbitrarily annotated as head-initial, is beyond the scope of this paper.

[6]An anonymous reviewer has remarked that the difference between Frank Bank and World Bank is arguably only a convention. This is certainly a valid point in general, but there is also some reason to consider differences between the structures, as codified in UD: while *World Bank* is without a doubt a kind of 'bank', the decision whether *Frank Bank* is a kind of 'Frank' or 'Bank' is more arbitrary. This becomes more crucial when nested compounds are considered, since multi-part names can be seen as truly flat, but compounds like *World Bank Federation* are recursive and right-headed.

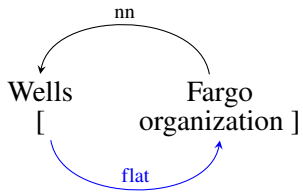

Figure 4: Analysis of 'Wells Fargo' (organization)

The semantic criterion distinguishing these two examples is that the dislocated node must be coreferential with a dependent of the verb ('them'='neighbors').[7] Because GUM has gold coreference annotations available, the preprocessing step again introduces a feature into the SD data which indicates a coreference ID for each coreferent nominal head, and nodes with the same coreference ID and syntactic head are changed from *dep* to *dislocated*.

Another type of information that may be seen as not purely syntactic is the presence of disfluencies. Though rare in written data, UD reserves a label for repairs in disfluencies or false starts, which can be used for both spoken and written data. The guidelines apply the label *reparandum* to the head of the 'aborted' part of the sentence, which is attached to the repair. The SD annotations in GUM follow the same structure, but apply the default label *dep*, meaning that the presence of the disfluency needs to be detected. This is accomplished in the preprocessing step by checking GUM's TEI XML annotations that denote all types of errors in the corpus with **<sic>** tags. Although these tags do not indicate the nature of the error or the repair, any occurrences of the *dep* label inside an error and governed from outside of it are converted into *reparandum*, as shown for the false start in Figure 5.

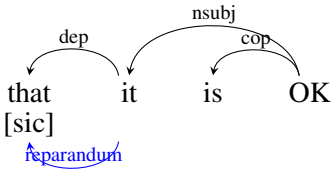

Figure 5: labeling errors as *reparandum*

We also use the **<sic>** annotations to create a feature in the MISC column defined by the CoNLL-U format with the value `Typo=Yes`, as used to denote errors in other UD treebanks. These are not necessarily always cases of repair, but also cases of unusual or non-standard grammatical constructions or even orthographic anomalies such as non-matching quotation marks, as in (3).

(3) so quite a few fans **<sic>known</sic>** about the "Mets Poet**<sic>'</sic>**

The same MISC column is also used to indicate whether tokens are followed by spaces using the feature `SpaceAfter=No`. The latter feature is also derived from the TEI annotations, where the presence of the tag **<w>** indicates multiple tokens spelled together as one orthographic word.

4 Evaluation

4.1 Experimental setup

We compare UD conversions from SD and constituent annotations in several scenarios on a total of 8,300 tokens, comprising just over 1,000 tokens from each genre in GUM, or about 10% of the corpus, for

[7] One anonymous reviewer has suggested that *dislocated* should be used for all fronted dependents, even if they are not realized a second time, citing a Japanese example from the UD guidelines. While we believe that marking fronting in general is interesting, and could perhaps be done using sublabels (e.g. *obj:front*), we feel that marking fronted English arguments, as in "him, I like" with *dislocated* is counter-intuitive, since it makes a verb such as 'like' appear to be missing an object. The practice in other English corpora, and specifically in EWT, has been to only mark *dislocated* in the presence of a second realization of the argument. The difference in the practice for Japanese may be due to the fact that in that language a second mention as a pronoun is usually omitted, and the closest equivalent of such a pronoun is therefore a zero-mention.

which we created manually checked gold UD parses. To evaluate constituent to UD ('C2UD') conversion accuracy, we created three constituent parsed versions of the same data using the Stanford Parser: one based on gold-tokenized plain text, one from data with gold POS tags, and the third, also parsed from gold POS tags, but then manually corrected for errors. The manually corrected constituent parses do not introduce empty categories such as PRO or traces, but do use function labels that may be critical for conversion, such as S-TPC (for fronted direct speech, common e.g. in fiction) and NP-VOC, NP-TMP and NP-ADV for vocative, temporal and other adverbial NPs.[8] C2UD conversion was carried out using CoreNLP 3.9.1, which uses built-in NER and heuristic time expression recognition, but is not completely up-to-date with the current UD standard. We therefore apply trivial renaming of labels where needed and two heuristic corrections: all coordinating conjunctions (labeled *cc*) are attached to the original target of the *conj* relation, so that they point right to left; and all nominal modifiers of verbs (labeled *nmod*) are re-labeled as *obl*.

In scoring correct conversion we focus on two metrics: attachment accuracy ignoring punctuation tokens (since punctuation is automatically attached using udapi, Popel et al. 2017, and errors are therefore by-products of other attachment errors), and label accuracy, including punctuation (since some punctuation symbols are occasionally used for non-punctuation functions). Because there are some differences in the label subtypes produced by CoreNLP and GUM (e.g. *obl:tmod*, *nmod:npmod*), we ignore subtypes for the evaluation and focus on main label types.

4.2 Results

Figure 6 shows boxplots for the range of error rates across documents from different genres in five scenarios (tokenwise micro-averaged global means are given in blue diamonds), each splits into two metrics: head and label accuracy.

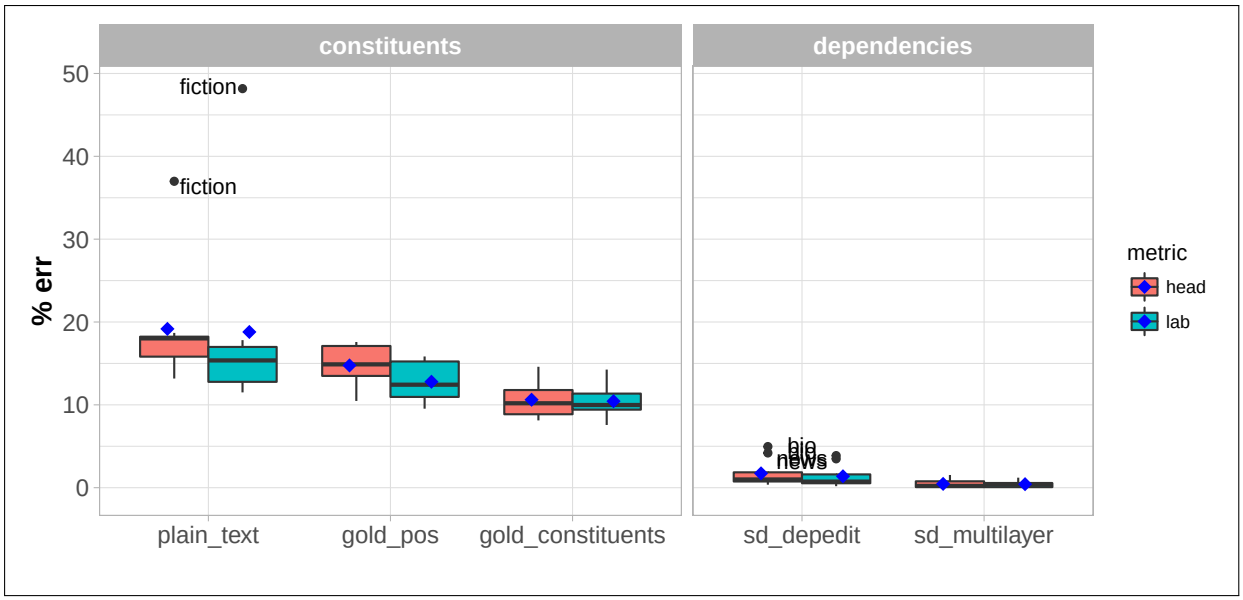

Figure 6: Error rates

In the best scenario, converting SD to UD with parallel multilayer information, conversion errors are very few, at 0.45%/0.42% of tokens (head/label errors). When multilayer annotations are removed, accuracy suffers somewhat, but is still rather good, with under 1.73%/1.38% errors. The more difficult genres for pure SD conversion are news and biographies, though only by a little: since these genres contain many multi-token proper names, correct conversion relies more on entity types, which cannot be recognized in the pure DepEdit conversion, but are available to the multilayer conversion.

[8]The data used for the evaluation, including different versions of constituent parses, is available at `https://github.com/gucorpling/GUM_UD_LAW2018`.

Comparing SD with constituent conversions, error rates become more substantial. Errors in the 'plain text' scenario are just under 20%; keeping in mind that the Stanford parser is trained on Wall Street Journal data, this is in line with previous results on parsing accuracy for out-of-domain constituent to dependency conversion (Choi and Palmer, 2010).

The not much better results for gold POS and gold constituents, by contrast, may seem surprising initially, since in general, constituents do identify the main argument structure relations, such as subjects and objects. However, a range of decisions cannot be made deterministically without semantic knowledge. Some of these might be avoided more reliably in datasets containing empty categories (traces, pro-forms) and more category sub-labels (e.g. PP-CLR, etc. see Bies et al. 1995), but the GUM constituents, even in their cleanest form, are based on CoreNLP constituent parses, which do not contain these.

Outliers in the 'plain text' scenario correspond to fiction texts, which frequently contained different Unicode quotation marks that are mistagged by CoreNLP. Gold POS tags remove the issue, as the 'gold pos' scenario shows. Nevertheless, even with gold constituents, mean error rates remain at 10.62%/10.44%. To understand the limitations of both constituent and SD to UD conversions, we examine some specific error patterns in the next section.

4.3 Error analysis

To understand what conversion errors need to be avoided, we first consider the difficulties in C2UD conversion. Table 2 shows the top 3 most frequent gold labels causing attachment and labeling errors for gold constituents, pure SD, and multilayer SD conversion.

scenario	**head errs**		**lab errs**	
C2UD	84	nsubj	130	obl
(gold)	82	nmod	74	nmod
	71	conj	62	conj
SD	37	flat	37	flat
(pure)	10	nmod	8	obl
	8	appos	7	nsubj
SD	8	compound	9	compound
(multi)	6	nmod	7	obl
	6	flat	6	nmod

Table 2: Top 3 gold labels showing head and label errors in three scenarios

In C2UD, even given gold constituents, many pure phrase labels are highly ambiguous with respect to their exact function. This is especially true for fronted NPs without function labels, which can be fronted arguments (*dislocated*, *obj*, *iobj*), a spatio-temporal adverbial (*advmod:npmod*), a vocative (*vocative*) and more. These are sometimes misidentified as subjects, leading to true gold subjects being misrecognized (objects are not as susceptible due to their position inside VPs). Conversely, the label *obl* is most often mislabeled, usually in cases where prepositional modifiers of nominal or adjectival predicates are not recognized and labeled *nmod*. In general, whenever phrases are extraposed, their attachment site cannot be predicted accurately in the absence of trace annotations, and these are most often labeled *nmod* and *obl*.

In third place, coordination is the next most problematic construction, due to the fact that PTB brackets do not explicitly mark coordination (except for Unlike Coordinate Phrases, labeled UCP). As a result, some non-standard but frequent types of coordination are missed, such as using '/' for 'or' (common in web data), 'et al.' (common in academic data) and unmarked coordination or lists using commas, which can look like appositions in constituent trees. All of these distinctions are represented directly in SD, which is conceptually much closer to UD, and thus these errors are virtually absent in the SD scenarios.

The errors in the Pure SD scenario are dominated by missing *flat* relations in proper names, due to the lack of entity recognition; guessing that all SD *nn* relations are UD *compound* is the safer choice. The confusion of *obl* and *nmod* features here as well, but is much less frequent, due to gold attachment data

in the SD parses which is usually trivial to convert to UD. Errors in appositions and subject relations are almost only by-products of incorrect name conversions, since the head token of the entire name is wrongly selected. In the Multilayer SD scenario, we see the over-generation errors in producing *compound* relations for non-person names – these are cases like 'Well Fargo', which should in fact be *flat* as well.

Additionally we note that the conversion from constituents is qualitatively missing some rare labels. These include cases that require the extra-syntactic knowledge described in section 3.2, such as *dislocated* and *reparandum*, but also the label *goeswith*, which indicates multiple tokens belonging to one 'word' but spelled apart, and the *vocative* label, which could hypothetically be guessed or derived directly if constituents include the NP-VOC subtype. While all of these labels represent rare phenomena, their exclusion from the constituent conversion output is problematic.

Finally we wish to point out one label that is currently not generated by any of our scenarios: the label *orphan*, which indicates promotion of a token to dominate the child of a missing coordinate parent. The construction, shown in Figure 7, is not directly expressible using SD relations and as such has been annotated somewhat unfaithfully by reference to the non-elliptical parent in the example.

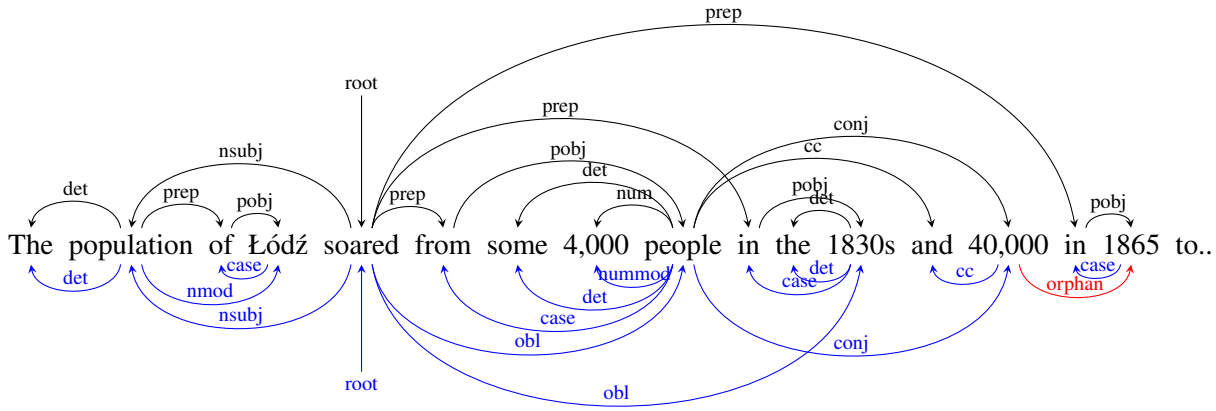

Figure 7: Example showing the *orphan* relation, not represented in SD

In the SD original (black edges), the population of Lodz is said to have soared from 4,000 people in the 1830s (two prepositional modifiers of 'soared'), and from 40,000 in 1865, to some other number. The inclusion of both the '1830s' and the '1865' as modifiers of 'soared' makes it seem as if both years apply at the same time. UD adds the relation *orphan* to express a second elliptical 'soared', which would have connected '40,000' and '1865' ("and [soared from] 40,000 in 1865..."). Though the UD solution seems clearly superior to the SD one, it is difficult to derive automatically without further annotations indicating the semantic structure, or using labels other than those found in SD.[9]

4.4 Comparison with other corpora

Aside from the accuracy of the conversion, we would like to suggest that there are some qualitative and quantitative differences between UD English data from 'native dependencies' and 'native constituents'. Qualitatively, some UD labels cannot be reliably produced via conversion, and are therefore absent in the initial C2UD result. This applies as noted above to the labels *dislocated* and *reparandum*, as well as *vocative* if conversion from NP-VOC is not used, though these labels can be reintroduced manually, as

[9]One reviewer has suggested that a better analysis of Figure 7 is to treat the two phrases after 'from' as a coordination, making the years part of the same constituents as the numbers of people, i.e.: "[from [[some 4000 people in the 1830s] and [40000 in 1865]]]...". However this solution incorrectly groups together the years and numbers, despite the fact that '4000 people in the 1830s' is not a constituent. Although the UD analysis with *orphan* is imperfect in not explicitly duplicating the node corresponding to 'soar', such an explicit analysis could be made using the optional Enhanced UD representation, which includes 'copy nodes'.

has been done in subsequent corrections to the EWT, for example.

Quantitatively, we note that non-projective dependencies, which are generally rare in English, are more frequent in SD2UD conversion than in C2UD. Table 3 shows frequencies for non-projective dependencies, excluding punctuation cases, across the entire EWT and GUM corpora in two scenarios: first, automatic C2UD conversion with CoreNLP is compared for both corpora. Then the current, partially manually corrected UD EWT V2.2 is compared with the multilayer conversion from SD for GUM, and the proportion of non-projectivity in the original gold SD data is given for comparison.

	C2UD	**UD V2.2 (corrected)**	
EWT	0.34%	0.46%	
	C2UD	**UD V2.2 (from SD multi)**	**original SD**
GUM	0.29%	0.79%	0.63%

Table 3: Non-projectivity in GUM and EWT.

The table shows that C2UD conversion creates less non-projectivity than human corrected or SD converted data, which is perhaps unsurprising. A more surprising result is that the manually corrected EWT contains substantially less non-projectivity than the SD2UD version of GUM. This could be due to genre differences, though the difference is rather substantial (almost double). If the numbers in EWT in fact under-represent the actual non-projectivity in the data, then this may be an indication that the less projective nature of the 'native constituents' EWT is shining through to the end result in the current UD version of the data. Finally we note that, at least for GUM, the conversion from gold SD to UD introduces further non-projectivity when compared to the original. A preliminary inspection of the constructions responsible for this suggests that coordinating conjunctions (the label *cc*) pointing backwards in UD instead of forwards in SD is responsible for the largest increase in cases of non-projectivity, but further study is needed to understand the extent and distribution of non-projective constructions generated by each scheme.

5 Discussion and outlook

The approach taken in this paper confirms that SD annotations are conceptually quite close to UD, making a purely rule-based conversion highly accurate. At the same time, we have shown that for some less frequent labels, information from annotation layers beyond the pure syntax tree is needed, and this reduces error rates from around 1.5% to closer to 0.4%. By contrast, conversion from constituent trees, even when these are manually checked, still results in around 10% errors (excluding punctuation).

An advantage of the present approach is the relative ease of the ability to change rules quickly as UD guidelines evolve: because the SD inventory is frozen, information that is derivable from the parse tree and further layers of annotation can be harnessed to produce the latest UD annotation scheme. It is also conceivable that retaining both SD and UD parses of the corpus can offer complementary information in some cases where UD collapses distinctions, e.g. between verbal modifiers labeled *vmod* in SD and other adverbial clauses labeled *advcl*.

One of the main limitations of the SD scheme with respect to producing the current UD standard is the lack of a function corresponding to *orphan*. This relation is also difficult for parsers to analyze correctly (see Schuster et al. 2018 for recent progress), meaning on the one hand that it is difficult to recognize automatically, and on the other, that it is desirable to include it in treebanks precisely in order to improve the availability of training data for such constructions.

In the future we would like to harness even more information from other layers in the corpus, both to enrich UD annotations with data in the MISC field and to validate annotation correctness. For example, using RST discourse parses available in GUM, we can draw on knowledge that certain clauses are *purpose* clauses to distinguish controlled to-infinitives (*xcomp*) from infinitival adverbial clauses (*advcl*). We are currently considering which other annotations can be used to enrich and improve the quality of UD corpora for which other concurrent annotations are available.

Acknowledgments

We would like to thank the reviewers, Nathan Schneider, and the UD community, for valuable comments on previous versions of this work, and the growing GUM annotation team for making their annotations available in the corpus – this resource could not have been created without their contributions. For the latest list of GUM contributors, please see the corpus website at `http://corpling.uis.georgetown.edu/gum/`.

References

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for Treebank II style. Penn Treebank Project. CIS Technical Report MS-CIS-95-06, University of Pennsylvania.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English Web Treebank. LDC2012T13, Linguistic Data Consortium, Philadelphia, PA.

Jinho D. Choi and Martha Palmer. 2010. Robust constituent-to-dependency conversion for English. In *Proceedings of the 9th International Workshop on Treebanks and Linguistic Theories (TLT 2010)*, pages 55–66, Tartu, Estonia.

Marie-Catherine de Marneffe and Christopher D. Manning. 2013. Stanford typed dependencies manual. Technical report, Stanford University.

Stefanie Dipper, Michael Götze, and Stavros Skopeteas. 2007. Information structure in cross-linguistic corpora: Annotation guidelines for phonology, morphology, syntax, semantics, and information structure. *Interdisciplinary Studies on Information Structure*, 7.

Roger Garside and Nicholas Smith. 1997. A hybrid grammatical tagger: CLAWS4. In Roger Garside, Geoffrey Leech, and Anthony McEnery, editors, *Corpus Annotation: Linguistic Information from Computer Text Corpora*, pages 102–121. Longman, London.

Geoffrey Leech, Tony McEnery, and Martin Weisser. 2003. SPAAC speech-act annotation scheme. Technical report, Lancaster University.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and Davide McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL 2014: System Demonstrations*, pages 55–60, Baltimore, MD.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Cristina Bosco, Gosse Bouma, Sam Bowman, Marie Candito, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Fabricio Chalub, Jinho Choi, Çağrı Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Linh Hà Mỹ, Dag Haug, Barbora Hladká, Petter Hohle, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Natalia Kotsyba, Simon Krek, Veronika Laippala, Lê Hồng, Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Hanna Nurmi, Stina Ojala, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh, Manuela Sanguinetti, Baiba Saulīte, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze,

Jonathan North Washington, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. 2017. Universal dependencies 2.0. Technical report, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Martin Popel, Zdenek Zabokrtský, and Martin Vojtek. 2017. Udapi: Universal API for Universal Dependencies. In *Universal Dependencies Workshop at NoDaLiDa 2017*, Gothenburg.

Beatrice Santorini. 1990. Part-of-speech tagging guidelines for the Penn Treebank project (3rd revision). Technical report, University of Pennsylvania, University of Pennsylvania.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In *Proceedings of LREC 2016*, pages 2371–2378, Portorož, Slovenia.

Sebastian Schuster, Joakim Nivre, and Christopher D. Manning. 2018. Sentences with gapping: Parsing and reconstructing elided predicates. In *Proceedings of NAACL 2018*, pages 1156–1168, New Orleans, LA.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel R. Bowman, Miriam Connor, John Bauery, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 2897–2904, Reykjavik, Iceland.

Amir Zeldes and Dan Simonson. 2016. Different flavors of GUM: Evaluating genre and sentence type effects on multilayer corpus annotation quality. In *Proceedings of LAW X The 10th Linguistic Annotation Workshop*, pages 68–78, Berlin.

Amir Zeldes. 2017. The GUM Corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.