Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks

Sebastian Schuster and Christopher D. Manning
Linguistics Department and Computer Science Department
Stanford University, Stanford, CA 94305
{sebschu,manning}@stanford.edu

Abstract

Many shallow natural language understanding tasks use dependency trees to extract relations between content words. However, strict surface-structure dependency trees tend to follow the linguistic structure of sentences too closely and frequently fail to provide direct relations between content words. To mitigate this problem, the original Stanford Dependencies representation also defines two dependency graph representations which contain additional and augmented relations that explicitly capture otherwise implicit relations between content words. In this paper, we revisit and extend these dependency graph representations in light of the recent Universal Dependencies (UD) initiative and provide a detailed account of an *enhanced* and an *enhanced*++ English UD representation. We further present a converter from constituency to *basic*, i.e., strict surface structure, UD trees, and a converter from *basic* UD trees to *enhanced* and *enhanced*++ English UD graphs. We release both converters as part of Stanford CoreNLP and the Stanford Parser.

Keywords: Universal Dependencies, semantic representation, treebank conversion

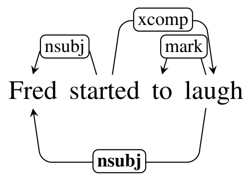
1. Introduction

Since its first version, the Stanford Dependencies (SD) representation (de Marneffe et al., 2006) has had the status of being both a syntactic and a shallow semantic representation. This dual status is also reflected in the usage of SD in natural language processing tasks which broadly fall into two categories. The first category is composed of tasks that require a syntactic tree such as source-side reordering for machine translation (e.g., Genzel (2010)) and sentence compression (Galanis and Androutsopoulos, 2010). For these tasks, a sound syntactic representation is more important than the relations between individual words.

The second and much larger category is composed of a wide range of shallow natural language understanding (NLU) tasks such as biomedical text mining (e.g., Airola et al. (2008)), open domain relation extraction (e.g., Mausam et al. (2012)), and unsupervised semantic parsing (Poon and Domingos, 2009). For these tasks, the relations between content words are more important than the overall tree structure.

Not surprisingly, we observe a similar divide if we look at which one of the three SD representations is being used for the individual downstream tasks. Most systems that require a syntactic representation use *basic* SD trees which are guaranteed to be a strict surface syntax tree. On the other hand, most systems that are concerned with the relations between content words use the *collapsed* or *CCprocessed* SD representations. These representations may be graphs instead of trees, and may contain additional and augmented relations that explicitly capture otherwise implicit relations between content words.

To illustrate one of the differences between these representations, consider the sentence *"Fred started to laugh"*. The *basic* SD representation of this sentence lacks a direct relation between the controlled verb *laugh* and its controller, *Fred*, while in the *CCprocessed* SD representation, this relation is made explicit with an additional subject edge.



The popularity of these extended representations suggests that their existence plays a major role in the popularity of Stanford Dependencies.

In recent years, there has been a lot of interest in extending Stanford Dependencies to other languages, including morphologically rich ones (McDonald et al., 2013; Tsarfaty, 2013; de Marneffe et al., 2014). These individual projects ultimately led to the Universal Dependencies (UD) initiative (Nivre et al., 2016) whose goal is to develop cross-linguistically consistent treebank annotations for as many languages as possible. While this project recognizes the status of dependency formalisms as semantic representations and based many design decisions on their impact on NLU tasks, the majority of efforts so far have focused on the development of the *basic* UD representation and the annotation of treebanks. Both de Marneffe et al. (2014) and Nivre et al. (2016) also mention an *enhanced* UD representation and acknowledge its usefulness but neither gives a detailed account of what such a representation should look like.

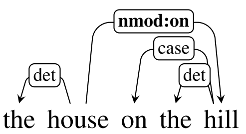
In this paper, we revisit and extend the *collapsed* and *CCprocessed* SD representations in light of the recent developments by the Universal Dependencies initiative. We provide a detailed account of an *enhanced* English UD representation and introduce the *enhanced*++ representation which we deem even better suited for many NLU tasks. Further, we describe our implementation of a converter from phrase-structure trees to *basic* UD trees, and a converter from *basic* to *enhanced* and *enhanced*++ English UD graphs. We release both tools as part of Stanford CoreNLP and the Stanford Parser.

2. The *enhanced* UD representation

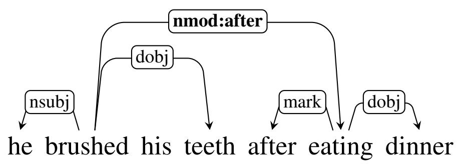
The *enhanced* English UD representation aims to make implicit relations between content words more explicit by adding relations and augmenting relation names. In the development of this representation, we adhered to the guidelines by Nivre et al. (2016) which state that an *enhanced* dependency graph may only contain additional dependencies or introduce additional language-specific relations.

As a result, *enhanced* UD graphs contain all the relations of the *basic* UD tree and the following additional relations.

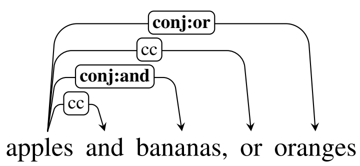
Augmented modifiers One major difference between the SD and UD representations is that the head of prepositional phrases (PP) is the prepositional complement and no longer the preposition itself. Therefore, there already exists a relation between the content word in the prepositional phrase and the word that is being modified by the PP in the *basic* UD representation, and there is no need for an additional relation in the *enhanced* representation. However, the *collapsed* SD graphs of sentences with PPs do not only contain additional relations, they also include the preposition in the relation name. This helps to disambiguate the type of modifier and further facilitates the extraction of relationships between content words, especially if a system incorporates dependency information using very simple methods such as by only considering paths between two nodes. For this reason, all nominal modifiers (`nmod`) in *enhanced* UD graphs also include the preposition in their relation name as exemplified in the following phrase.



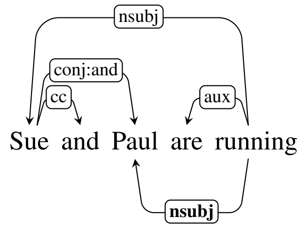
The same is true for more complex PPs which are either analyzed as adverbial clause modifiers (`advcl`) or as adjectival clause modifiers (`acl`) in UD. If an adverbial clause or an adjectival clause is introduced by a marker (`mark`), we add the marker to the relation name.



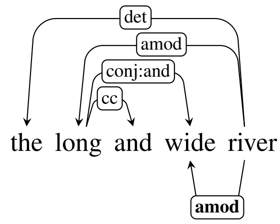
Augmented conjuncts In a similar manner, *enhanced* UD graphs also contain conjunct relations that are augmented with their coordinating conjunction. This makes the type of coordination between two phrases more explicit which is particularly useful in phrases with multiple coordinating conjunctions, such as the following phrase.



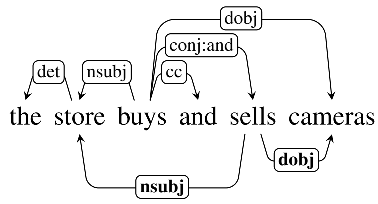
Propagated governors and dependents In *basic* UD trees of clauses with conjoined phrases, only the first conjunct has explicit relations to the governor and the dependents of the conjoined phrase. In the *enhanced* UD graph, the implicit relations of the other conjuncts are made explicit with additional relations. In the case of conjoined noun phrases, each noun phrase becomes an argument of the head of the first conjunct, e.g., it becomes the subject of the main verb as in the following example.



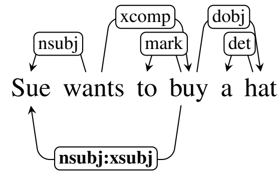
In the case of conjoined adjectival phrases, each adjective becomes a modifier of the head of the first conjunct.



In the case of conjoined verbs, the arguments of the first verb, e.g., the subject and the direct object, also become the arguments of the other verbs.

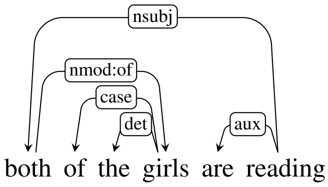


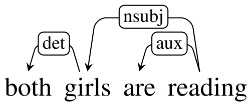
Subjects of controlled verbs *Basic* UD graphs of sentences that contain a controlled embedded verb lack a direct relation between the controlled verb and the controller. Therefore, the *enhanced* UD graphs contain a special controlling subject (`nsubj:xsubj`) relation between the embedded verb and the controller.



3. The *enhanced*++ UD representation

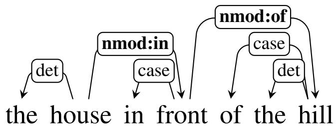
The *enhanced* representation provides reasonable analyses for most English sentences. However, there are some constructions in English that lead to an analysis which is suboptimal for many NLU systems that try to extract relationships between entities, such as open domain relation extraction (e.g., Mausam et al. (2012)), or extracting relationships between objects in image descriptions (Schuster et al., 2015). One set of problematic constructions involves partitive noun phrases such as *both of the girls* in which *both of the* acts semantically as a quantificational determiner. In the *basic* UD representation, however, *both* is the head of such a partitive phrase while the semantically very similar phrase *both girls* is headed by *girls*:





Considering that many relation extraction systems use simple dependency tree patterns to extract entities and their relations, these different analyses are clearly problematic. In the first phrase, the determiner *both* appears to be the subject while in the second phrase, *girls* is the subject, but ideally both phrases would be analyzed in a similar way. However, in order to obtain a similar analysis for both phrases, we would have to change the structure of the *basic* dependency trees, which is not allowed according to the guidelines for *enhanced* dependency graphs.

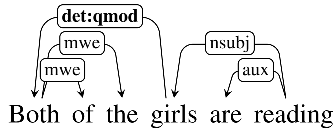
Another set of problematic constructions involves multi-word prepositions such as *in front of*. As illustrated in the following tree, the *enhanced* representation of "*a house in front of the hill*" contains a relation between *house* and *front*, and a relation between *front* and *hill*.



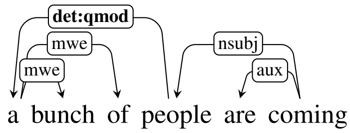
But for most tasks, the relation between *house* and *hill* is going to be more relevant. This relation could be made explicit by adding another relation between *house* and *hill*, but as we are not allowed to delete any relations from the *basic* representation, we would encode some information twice.

These two issues should illustrate that there exist several phenomena for which both the *basic* UD representation and the *enhanced* UD representation provide suboptimal analyses. For this reason, we argue for another representation which allows for the deletion of relations from the *basic* UD tree, as this gives us more flexibility in analyzing several constructions in English, including the ones mentioned above. We therefore introduce *enhanced++* UD graphs and propose different analyses for the following common phenomena in English.

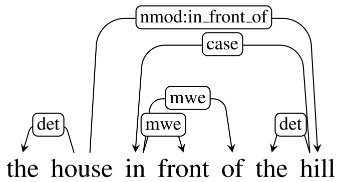
Partitives and light noun constructions For the analysis of partitive noun phrases such as *both of the girls*, we follow Barwise and Cooper (1981) and Keenan and Stavi (1986) and treat the first part of the phrase as a quantificational determiner. We promote the semantically salient noun phrase, e.g., *girls* in our example, to be the head of the partitive and we analyze the quantificational determiner as a flat multi-word expression that is headed by its first word. In order to mark that these quantificational determiners are not regular determiners, we attach them using the special relation quantificational modifier (`det:qmod`).



Light noun constructions (Simone and Masini, 2014) such as *a panel of experts* or *a bunch of people* pose similar challenges because the light nouns are the head of these phrases in the corresponding *basic* UD trees. However, just like the partitives, the second noun phrase tends to be the semantically salient one in these constructions while the first part of the phrase again serves as a quantificational determiner. We therefore analyze light noun constructions exactly like partitives as illustrated in the following example.[1]



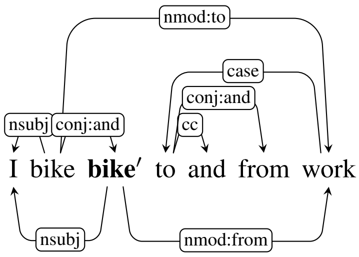
Multi-word prepositions As mentioned above, multi-word prepositions such as *in front of* tend to obscure the relation between two content words. While the *basic* UD representation analyzes some multi-word expressions with function words, e.g., *due to*, using a special `mwe` relation, the set of these expressions is very limited and does not include many multi-word prepositions. To introduce a direct relation between content words in the *enhanced++* UD representation, we also analyze these multi-word prepositions as flat multi-word expressions headed by the first word, and we attach the head of the phrase to the following noun phrase.



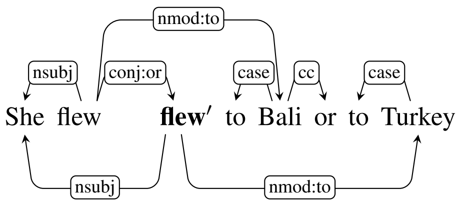
Conjoined prepositions and prepositional phrases Clauses that contain conjoined prepositions such as *"I bike to and from work"* also pose some challenges. Ideally, the UD graph should encode that there is an `nmod:to` as well as an `nmod:from` relation between *bike* and *work*. Further, we also want to encode that *bike to work* and *bike from work* are conjoined by *and*. In order to be able to represent all of this information, the *CCprocessed* SD representation introduced copy nodes which we adapt in the *enhanced++* representation. The analysis of this example then contains

[1]The special treatment of light noun constructions also raises the question of how we should treat light verb constructions (LVCs) (Jespersen, 1954) such as *to make a decision*. In many contexts, these constructions have a very similar meaning as a semantically strong verb, e.g., *decide*, and ideally sentences would be analyzed in similar ways independent of whether they contain a LVC or a semantically strong verb. However, unlike in the case of light noun constructions, we cannot achieve this goal by solely adding or removing edges, and instead would have to modify surface tokens, e.g., turning *decision* into *decide*. Because of this and other issues concerning LVCs, we currently do not analyze LVCs in a special way.

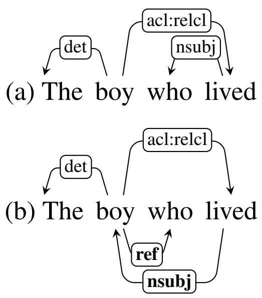
a copy of *bike*, namely *bike′*. This copy node is attached to the original node as a conjunct resulting in the following UD graph.



Note that this graph contains the same relations between content words as the *basic* UD tree of the clause *"I bike to work and I bike from work"*.
Similar complexities arise with clauses that contain a conjoined prepositional phrase such as *"She flew to Bali or to Turkey"*. Again, the UD graph should encode that there is an *nmod:to* relation between *flew* and *Bali* and another *nmod:to* relation between *flew* and *Turkey* and at the same time it should encode that these two relations are conjoined by *or*. For these reasons, we also analyze such clauses with copy nodes.



Relative pronouns We also analyze relative pronouns differently in the *enhanced*++ representation as compared to the *basic* UD representation. Similar to the *collapsed* SD representation, the referent of the pronoun is directly attached to the governor of the pronoun. Further, we attach the relative pronoun to its referent with a referent (`ref`) relation.
The following example illustrates the differences between *basic* UD and *enhanced*++ UD graphs which contain relative clauses. In the *basic* UD tree (a), the head of the relative pronoun *who* is *lived* and there is no direct relation indicating that *boy* is an argument of *lived*. In the *enhanced*++ representation (b), on the other hand, *boy* is the subject of *lived*, and *who* is attached to its referent, *boy*.



4. Generating dependency trees and graphs

There exist multiple ways to obtain trees and graphs in the various UD representations for a given sentence. The *basic* UD trees can be either generated directly by using a dependency parser or by using a constituency parser followed by a converter from phrase structure to dependency trees. The *enhanced* and *enhanced*++ dependency graphs can be obtained by post-processing *basic* dependency trees.
In the following two sections, we describe our converter from phrase structure to *basic* English UD trees and how to obtain *enhanced* and *enhanced*++ dependency graphs from *basic* English UD trees.

4.1. Converting to *basic* dependencies

Our converter from phrase structure to dependency trees is based on the Stanford Dependencies converter (de Marneffe et al., 2006) which we updated according to the English Universal Dependencies guidelines.
To determine the structure of the dependency tree, we use a semantic head finder which operates similarly to the Collins head finder (Collins, 1999). For each constituent type, we define a set of rules that determine from which of its children the constituent inherits its head; terminal nodes have their surface token as head. These rules are mostly conditions on the constituent type of the child but unlike the classic Collins head finder rules, some of them also take surface tokens into account which is necessary for distinguishing between main verbs and auxiliaries. We traverse the constituency tree in depth-first order and use these rules to obtain and store the head of each constituent, resulting in a tree in which every node has exactly one surface token as head. The head of each token is then simply the head of its lowest ancestor whose head is not the token itself.
To determine the relation types, we define for each grammatical relation a set of tree patterns in the form of `tregex` expressions (Levy and Galen, 2006). For each head-dependent pair we try to find a pattern that matches the subtree rooted at the lowest common ancestor of the head and the dependent. If such a pattern exists, we assign its corresponding grammatical relation to the head-dependent pair. In the rare cases where no pattern matches, we assign the most general relation *dep*.
This procedure allows us to obtain correct dependency trees in most cases. Two phenomena, however, require additional consideration. First, as previously mentioned, the UD representation defines several multi-word expressions with function words that behave like a single word such as *because of* or *in case*. Extracting the correct structure for these expressions is often challenging because many of these expressions are not a constituent according to the Penn Treebank annotation guidelines (Marcus et al., 1993). We resolve this issue by preprocessing all phrase structure trees that contain multi-word expressions such that the entire expression forms a constituent.
Second, the outlined procedure often attaches wh-words in questions to the wrong head. For a question such as *What does Peter seem to have?*, our procedure would attach *what* to the head of the matrix clause, *seem*, instead of the head of the embedded clause, *have*. If we were only concerned with converting manually annotated treebanks, we could resolve these ambiguities by making use of the indexed empty nodes in the phrase structure trees, as proposed by Choi and Palmer (2012). However, the output of most constituency parsers does not contain these empty nodes. Therefore, we try to solve this issue by considering the selectional restric-

tions of the verb in the matrix clause and if the attachment of the wh-word violates these restrictions we try to reattach it to the head of the embedded clause.

4.2. Converting to *enhanced* and *enhanced++* dependencies

For the conversion from *basic* dependencies to *enhanced* and *enhanced++* dependencies we mainly rely on dependency tree patterns in the form of `Semgrex` expressions (Chambers et al., 2007). `Semgrex` expressions allow one to match subgraphs of a dependency graph based on properties of the nodes and their relations.

For most of the enhancements, we search for syntactic patterns and then either modify the relation name or introduce new relations. The multi-word prepositions and the quantificational constructions, however, do not match any distinct dependency patterns. For these modifications, we rely on lists of specific expressions and modify the graph structure whenever we encounter one of them.

Our converter generates *enhanced* and *enhanced++* dependency graphs as described in the previous sections, with one exception. Currently, we don't propagate object or nominal modifier relations in clauses with conjoined verb phrases such as "*the store buys and sells cameras*". The reason for this is that there are also many cases such as "*she was reading or watching a movie*" where it would be wrong to add these relations and there are no syntactic cues that would allow us to distinguish these cases. Nyblom et al. (2013) successfully used a machine learning approach to solve this problem for Finnish but as there currently exists no corpus annotated with *enhanced* English UD graphs, we leave this to future work.

4.3. Evaluation

We evaluate our basic converter against the manually checked English UD treebank v1.1 (Nivre et al., 2015) which contains annotations for all sentences in the EWT corpus (English Web Treebank, Linguistic Data Consortium release LDC2012T13). We convert all phrase structure trees in the EWT corpus to *basic* UD trees and compare the output with the manually checked trees using the official CoNLL Shared Task evaluation script.

The results of this evaluation are presented in Table 1. These results indicate that our converter is able to convert phrase structure trees from a variety of genres to *basic* UD trees at high accuracy. If we compare the performance of the converter across the individual genres, we can see that the converter performs best on sentences from weblogs and newsgroups and slightly worse on sentences from emails, from an online question-answering site, and from online business reviews. A qualitative error analysis showed that the main reason for the small drop in performance on the question-answer and review corpora is that these corpora contain a lot of ungrammatical sentences written by non-native English speakers. The main reason for the lower performance of the converter on the email corpus is that this corpus contains a lot of corporate email signatures whose corresponding phrase structure trees consist of a single flat fragment from which it is very hard to extract properly structured dependency trees.

Genre	LAS	UAS	Accuracy
Question-answers	92.0	95.4	93.7
Email	91.4	95.8	92.7
Newsgroups	93.1	96.8	94.0
Business reviews	92.5	95.9	93.9
Weblogs	94.5	97.1	95.7
Entire corpus	92.6	96.1	93.9

Table 1: Labeled attachment score (LAS), unlabeled attachment score (UAS), and accuracy of the converter from phrase structure trees to *basic* English UD trees on the individual genres of the English Web Treebank corpus. We use the manually corrected English UD corpus v1.1 (Nivre et al., 2015) as a gold standard.

4.4. Applications

We believe there are two main applications of our converters. First, our *basic* UD converter can be used to automatically convert existing treebanks of phrase structure trees to treebanks of UD trees for training dependency parsers. For example, we have successfully converted the entire Penn Treebank (Marcus et al., 1993) to train models for a neural network dependency parser (Chen and Manning, 2014).

Second, our converters can be used either in combination with a constituency parser or a dependency parser to obtain UD graphs for any sentence which can then be utilized in downstream NLU tasks. Schuster et al. (2015), for example, used a preliminary version of the converter to obtain *enhanced++* UD graphs from constituency trees. This system uses the UD graphs as input for a parser from image descriptions to a scene representation that captures relationships between objects in a visual scene. Further, the open domain information extractor in CoreNLP (Angeli et al., 2015) also already uses UD graphs to extract relations between entities.

5. Comparison to AMR

Representing the meaning of sentences as directed graphs has a long tradition in computational linguistics, which goes back to at least Shieber (1984). One graph-based semantic representation that received significant attention in recent years is the Abstract Meaning Representation (AMR) (Banarescu et al., 2013). AMR also encodes sentences as directed graphs but compared to UD graphs, it aims to abstract further away from the surface form of sentences. To achieve this goal, it encodes sentences using PropBank framesets (Palmer et al., 2005) and approximately 100 fixed relations. This makes AMR a deeper and more canonicalized semantic representation as compared to UD graphs. While these are obviously desirable properties, we nevertheless believe that our representation has some advantages over AMR, especially when it is being used in shallow natural language understanding tasks.

In terms of expressivity, UD graphs have the advantage that they encode the meaning of sentences in terms of relations between surface form tokens and they are therefore as expressive as natural language. The expressivity of AMR, on the other hand, is constrained by the finite set of PropBank framesets. This is particularly problematic for neologisms such as *to venmo* for which no corresponding PropBank framesets exist.

Further, we are using existing resources of languages to disambiguate argument types, e.g., by including the preposition in the relation name of nominal modifiers, which avoids many hard labeling decisions for parsers as well as for human annotators. While this approach is occasionally too simplistic as our representation does not, for example, distinguish between comitative *with* and instrumental *with*, it nevertheless sufficiently disambiguates argument types for most domain-specific NLU tasks. AMR, on the other hand, requires the labeling of every argument with a semantic role which is - apart from labeling agents and patients - a very hard task.

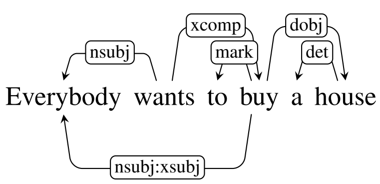
Another appeal of UD graphs is their simplicity. A lot of the frequent relations, such as *nominal subject* or *object* are very intuitive which makes them easily interpretable by non-experts. Compared to UD graphs, AMR graphs are more complex and require more explanation even if one is only interested in extracting simple relations such as subject-verb-object triplets.

Finally, from a practical point of view, sentences can be parsed to UD graphs with much higher accuracy than to AMR graphs. While the existence of high-performing parsers clearly should not be the main desideratum in the design of a semantic representation, this aspect plays ultimately an important role in the usefulness of a representation.

6. Limitations

As explained above, most of the additional relations in the *enhanced* and *enhanced++* representations can be added with syntactic rules. This purely syntactic approach tends to work well in practice, and has the appeal of being very simple and easily comprehensible. However, in some cases this approach leads to UD graphs that encode a different meaning than the original sentence, which can be problematic in downstream NLU tasks.

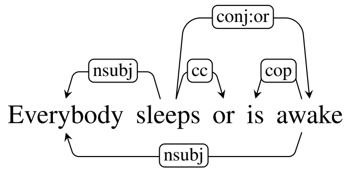
One issue concerns clauses with generalized quantifiers and controlled verbs, such as the following sentence[2]



The issue with this UD graph is that it does not encode the meaning of the original sentence but instead encodes approximately the meaning of the sentence *"Everybody wants that everybody buys a house"*. In order to preserve the original meaning and to encode the relation between the controlled verb and its subject we would have to introduce variables and consequently abandon one of our core principles, namely that we encode the meaning of a sentence in terms of relations between surface form tokens.

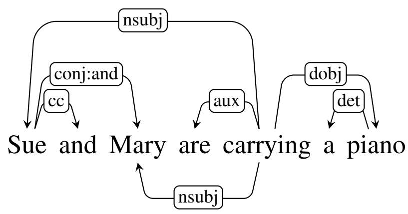
[2]Thanks to Christopher Potts for pointing out this issue.

A second issue concerns the propagation of dependents. Note that in the case of conjoined verbs or verb phrases, we are effectively performing a reverse conjunction reduction which can lead to problematic analyses in combination with generalized quantifiers. For example, consider the sentence *"Everybody sleeps or is awake"* with the following *enhanced* UD graph.



The issue with this UD graph is that it approximately encodes the meaning of the sentence *"Everybody sleeps or everybody is awake"* which again differs from the meaning of the original sentence.

Lastly, another issue concerns sentences with conjoined subjects, such as *"Sue and Mary are carrying a piano"*.



Unlike in the previous two examples, the issue with this UD graph is not that it encodes a different meaning than the original sentence, but rather that this representation of conjoined subjects favors a distributive interpretation. Ideally, we would propagate the subject relation only when it is clear that a distributive interpretation is intended, but as the question whether a conjoined subject should be interpreted distributively or collectively also depends on world knowledge and the context, we are not able to make this distinction based on the information that is contained in a single sentence.

7. Conclusion

In this paper, we presented the first detailed account of an *enhanced* English Universal Dependencies representation. We further argued for additional modifications of the tree structure to facilitate extracting relations between content words and described these modifications as part of the *enhanced++* representation. Finally, we described how both of these representations can be automatically generated from phrase structure trees or *basic* dependency trees with high accuracy.

8. Acknowledgements

This work greatly benefitted from many improvements to the original Stanford Dependencies converter by Marie-Catherine de Marneffe and John Bauer, which we gratefully acknowledge. We also thank Natalia Silveira and Timothy Dozat for helpful discussions, and the entire Universal Dependencies community for developing the guidelines and their valuable input. This work was supported in part by a gift from IPSoft, Inc. The first author is also supported by a Goodan Family Graduate Fellowship.

9. Bibliographical References

Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., and Salakoski, T. (2008). All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, 9(Suppl 11):S2.

Angeli, G., Premkumar, M. J., and Manning, C. D. (2015). Leveraging linguistic structure for open domain information extraction. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL.*

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*.

Barwise, J. and Cooper, R. (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219.

Chambers, N., Cer, D., Grenager, T., Hall, D., Kiddon, C., MacCartney, B., de Marneffe, M.-C., Ramage, D., Yeh, E., and Manning, C. D. (2007). Learning alignments and leveraging natural logic. In *ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Chen, D. and Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Choi, J. D. and Palmer, M. (2012). Guidelines for the Clear style constituent to dependency conversion. Technical report, Technical Report 01-12, University of Colorado at Boulder.

Collins, M. (1999). *Head-driven statistical models for natural language processing*. Ph.D. thesis, University of Pennsylvania.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006)*.

de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. D. (2014). Universal Stanford Dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Galanis, D. and Androutsopoulos, I. (2010). An extractive supervised two-stage method for sentence compression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Genzel, D. (2010). Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*.

Jespersen, O. (1954). *A modern English grammar*, volume 3. George Allen & Unwin Limited.

Keenan, E. L. and Stavi, J. (1986). A semantic characterization of natural language determiners. *Linguistics and Philosophy*, 9(3):253–326.

Levy, R. and Galen, A. (2006). Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *5th International Conference on Language Resources and Evaluation (LREC-2006)*.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2).

Mausam, Schmitz, M., Bart, R., Soderland, S., and Etzioni, O. (2012). Open language learning for information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*.

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Bertomeu Castelló, N., and Lee, J. (2013). Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nivre, J., Agić, Ž., Aranzabe, M. J., Atutxa, A., Bosco, C., Choi, J., de Marneffe, M.-C., Dozat, T., Farkas, R., Foster, J., Ginter, F., Goenaga, I., Gojenola, K., Goldberg, Y., Hajič, J., Johannsen, A. T., Kanerva, J., Kuokkala, J., Laippala, V., Lenci, A., Lindén, K., Ljubešić, N., Lynn, T., Manning, C. D., Martínez, H. A., McDonald, R., Missilä, A., Montemagni, S., Nurmi, H., Osenova, P., Petrov, S., Piitulainen, J., Plank, B., Prokopidis, P., Pyysalo, S., Seeker, W., Seraji, M., Silveira, N., Simi, M., Simov, K., Smith, A., Tsarfaty, R., Vincze, V., and Zeman, D. (2015). Universal Dependencies 1.1. http://hdl.handle.net/11234/LRT-1478.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC-2016)*.

Nyblom, J., Kohonen, S., Haverinen, K., Salakoski, T., and Ginter, F. (2013). Predicting conjunct propagation and other extended Stanford Dependencies. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Poon, H. and Domingos, P. (2009). Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Schuster, S., Krishna, R., Chang, A., Fei-Fei, L., and Manning, C. D. (2015). Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the Fourth Workshop on Vision and Language (VL15)*.

Shieber, S. M. (1984). The design of a computer language for linguistic information. In *Proceedings of the 10th International Conference on Computational Linguistics*.

Simone, R. and Masini, F. (2014). On light nouns. *Word Classes: Nature, typology and representations*, 332:51.

Tsarfaty, R. (2013). A unified morpho-syntactic scheme of Stanford Dependencies. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*.