Can predicate-argument relationships be extracted from UD trees?

Adam Ek **Jean-Philippe Bernardy** **Stergios Chatzikyriakidis**
Centre for Linguistic Theory and Studies in Probability
Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
{adam.ek,jean-philippe.bernardy,stergios.chatzikyriakidis}@gu.se

Abstract

In this paper we investigate the possibility of extracting predicate-argument relations from UD trees (and enhanced UD graphs). Concretely, we apply UD parsers on an English question answering/semantic-role labeling data set (FitzGerald et al., 2018) and check if the annotations reflect the relations in the resulting parse trees, using a small number of rules to extract this information. We find that 79.1% of the argument-predicate pairs can be found in this way, on the basis of Udify (Kondratyuk and Straka, 2019). Error analysis reveals that half of the error cases are attributable to shortcomings in the dataset. The remaining errors are mostly due to predicate-argument relations not being extractible algorithmically from the UD trees (requiring semantic reasoning to be resolved). The parser itself is only responsible for a small portion of errors. Our analysis suggests a number of improvements to the UD annotation schema: we propose to enhance the schema in four ways, in order to capture argument-predicate relations. Additionally, we propose improvements regarding data collection for question answering/semantic-role labeling data.

1 Introduction

Universal Dependencies (UD), can be seen as a compromise, a balancing act between six principles, referred to as Manning's law (Nivre et al., 2016):

1. UD needs to be satisfactory for analysis of individual languages

2. UD needs to be good for linguistic typology

3. UD must be suitable for rapid, consistent annotation

4. UD must be suitable for computer parsing with high accuracy

5. UD must be easily comprehended and used by a non-linguist

6. UD must provide good support for downstream language understanding tasks

Support for natural language understanding downstream tasks in the UD schema has been shown in a number of studies including event extraction, negation scope detection and opinion analysis (Fares et al., 2018), information extraction (Angeli et al., 2015), image retrieval (Schuster et al., 2015), question-answering (Reddy et al., 2017), and Natural Language Inference (Mishra et al., 2020), among many others.

However, certain syntactic dependencies relevant to semantics are not included in the original formulation of UD. For example, a word may be the subject of two conjoined verbs, but in UD the subject is only connected to one of the verbs. To discover that the word is the subject of two verbs it has to be inferred from the conjunction. However, this creates unnecessary burdens for models using the UD schema. The *enhanced UD schema* (EUD) (Schuster and Manning, 2016) includes such edges, with the aim to make semantics more explicit. Recently there has been a surge of interest and development of EUD, spurred on by its applicability on semantic downstream tasks such as information extraction (Tiktinsky et al., 2020; Sun et al., 2020). Research into EUD has also be facilitated recently by two shared tasks on EUD parsing (Bouma et al., 2020, 2021), which has resulted in a mix of machine learning and rule-based approaches for producing EUD graphs. We come back to an evaluation of the EUD schema in Section 5.1.

The support provided by UD w.r.t. downstream NLU tasks raises the question of how much "semantics" UD actually contains, or better put, how much semantic reasoning can one perform by using just the information provided by UD. This is also related to the question of whether UD dependencies

Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop, pages 46–55
November, 2021. ©2021 Association for Computational Linguistics

should be seen as semantic, syntactic, or maybe something between the two. To some extent all three possibilities have been considered. One way to approach this question is to check the amount of semantic knowledge that UD exhibits, explicitly or implicitly, in relation to specific semantic tasks or features. Silveira (2016) argues that the way to see UD is as a representation "for" semantics, not "of" semantics. Under this view, UD can be seen as a kind of scaffolding where some proper semantic backbone will be built upon. Again, however, this begs the question of the nature of the scaffolding. Silveira (2016) claims that UD has implicit semantic role information and also shows that their enhanced version, which, as they argue, mirror semantic relations more closely, perform better than normal UD in an event extraction task involving a model that extracts dependency features from different parses. Previous research has shown the opposite to be the case, i.e. UD performing better than the enhanced version in this task Miwa et al. (2010a,b); Buyko and Hahn (2010), even though these pieces of work are not directly tested on enhanced UD, but on previous related efforts to expand basic UD (Silveira, 2016). UD has been also criticized by researchers working in Theoretical Linguistics (Osborne and Gerdes, 2019). According to them, UD fails to observe Manning's first desideratum because "UD annotation choices are not satisfactory on linguistic analysis grounds because they result from a mixture of semantic and syntactic criteria". Lastly, one could argue that approaches that attempt to combine UD with an explicit logical semantics interface implicitly assume that UD is syntactic and/or missing crucial semantic information.

In this paper, we propose a way to test the semantic capabilities of UD *parsers* for English by using their output to infer answers in a Question-Answering task. More precisely, what we want to investigate is the question of whether predicate-argument relations are correctly captured by UD parsers. We believe that this is an important question to be posed, because, if this is the case and there is enough ground/scaffolding, then a more fine-grained semantic representation may be build on top of UD (for example, some correspondence between UD syntactic trees and logical semantics). A related question is to what extent enhanced dependencies are better, if at all, in precisely encoding predicate-argument relations.

2 Dataset

We perform experiments on the question-answering/semantic-role-labeling dataset of (FitzGerald et al., 2018), which is based on the work of (He et al., 2015), simply referred to as "QA-SRL" below. The rationale is that, in the QA-SRL dataset, question-answers pairs are directly concerning predicate-argument structures. Each question has a passage which it refers to. For example, the dataset might contain the passage "UN published a report" together with the question "What did something publish?". The answers are provided by annotators selecting a contiguous span of text in the passage which answers the question, in this case the object "a report".

The dataset contains passages from 3 domains in English: Wikipedia, Wikinews and science, with questions and answers generated by crowdsourcing. For each verbal predicate in the passage, questions about one of the arguments are constructed by the annotators using question templates. In total the dataset contain 265156 valid questions over 76397 passages. The QA-SRL dataset also contains an automatically generated dataset. However, we have not included this part and only consider the crowd-sourced part.

3 Task and Method

The most obvious way to test whether UD parsers can correctly identify the semantic arguments of verbs would be to map the form of a QA-SRL question to an UD role, then retrieve the subtree of the argument from the UD tree and check if it matches the human annotations.

Unfortunately it is not easy to map the argument types of the QA-SRL dataset to UD roles. One difficulty is the mismatch of passive and active voice between questions and answers. Another problem is that the non-subject UD roles (obj/obl/advcl/etc) are in n-to-n correspondence with the QA-SRL argument types (locations, time, etc). Converting these relationship to a functional mapping would require the use of some statistical model to extract these features from the sentence. Using a statistical model would make unclear whether it is UD that captures argument-predicate relationships, or the model. Thus, to keep the method simple we resort to checking if the UD trees obtained from a parser contains the annotated QA-SRL argument. To avoid the question of *which* semantic role should be extracted, we check if *any* of the children of the verb matches the answer. We make two further amendments to the task: 1. we enhance UD trees with EUD arcs and 2. we check for arguments in the parent position.

The second amendment helps with cases when the sentence has the form of a copula or when the verb plays the role of adjectival phrase. For example, given the passage "Paleontologists are interested in fossils" and the question "Who is interested in something?", then one should be able to recover "Paleontologists" as an argument. However, in the UD tree, "Paleontologists" is the parent of "interested". Likewise, given "The observed animals were tortoises." and the question "What was observed?" should point to "animals"; which is the parent of "observed" in the UD tree.

The first amendment is to use the EUD schema rather than plain UD. While the state-of-the-art UD parsers do not provide this information, it is possible to automatically add most EUD edges using a number of rules (Silveira, 2016; Ek and Bernardy, 2020). Thus our pipeline consists in first running a plain UD parser, we test both the Stanza parser (Qi et al., 2020) and the Udify parser (Kondratyuk and Straka, 2019), and then we apply the following enhancements to the UD trees, using the system developed in (Ek and Bernardy, 2020):

1. Propagation of incoming dependencies to conjuncts;
2. Propagation of outgoing dependencies from conjuncts;
3. Propagation of subject relations for direct control and raising constructions;
4. Addition of co-reference arcs in relative clause constructions

To recapitulate, after adding enhanced edges for each question in the test set, we proceed to:

1. Find the verb index relevant to the question. Generally this information is given by the QA-SRL data. In rare cases some adjustments need to be made, for example if the parser counted words differently than the dataset we adjust the verb index accordingly;
2. Collect all possible arguments according to the EUD graph;
3. Extract the constituent for each argument by following the child edges;
4. Normalize the text of each constituent by removing punctuation, leading prepositions, and determiners. Indeed, the annotations are inconsistent regarding whether prepositions and determiners should be part of the argument or not;
5. If any of the gold answers match any of the arguments retrieved, we consider the argument retrieval a success

4 Results and Analysis

In this section we present the results obtained from extracting predicate-argument relations, and provide an analysis of the errors observed.

4.1 Baseline

As a side experiment, we have attempted to find if the argument can be found *anywhere* as a constituent in the UD parse tree.

Model	Upper bound
Udify	98.9%
Stanza	98.6%

Table 1: Dependency parsers upper bound performance.

Table 1 shows that in 98.9 and 98.6 of the cases, it is possible to extract the semantic arguments from the syntactic structure by finding an appropriate root of the tree. Thus, the above numbers place a theoretical upper bound on the method, as the accuracy that we could achieve if arguments were always correctly attached to their predicate. This means that the above numbers provide a sanity check for the approach: in 98.9% of the cases, the gold correspond to something which Udify has identified *somewhere* in the sentence.

4.2 Extracting predicate-argument relations

In Table 2 we report the accuracy for both parsers, with and without the applying the enhancements described in Section 3. The results show a clear superiority for Udify, which is more than 4 percentage points above Stanza in both configurations.

Parser	Plain UD	EUD
Udify	0.683	0.791
Stanza	0.722	0.744

Table 2: Accuracy of UD trees with and without enhancements using the Udify and Stanza parsers.

Taking into account enhancement edges gives a large benefit to Udify parser, and a small benefit to Stanza.

To get a better sense of where the errors are coming from, we have performed manual analysis as follows. Focusing on the best performing configuration (Udify with enhanced dependencies), we picked 100 test cases at random, and, by manual inspection, we determined if the error is imputable to either the parser, the dataset or the method. Our classification criteria are as follows:

Parser If the used UD parser produced a wrong parse tree.

Dataset If either the passage or the question is incorrect, either syntactically or semantically; or if the annotations do not contain the answer according to the question and passage.

Method If both the dataset and the parse tree are correct, but the argument is not related to the verb in the UD tree.

We found the following results: out of 100 cases, 49 errors were attributable to the dataset, 13 to the parser and 38 to the method. In terms of percentage points of lost accuracy, this means that 10.2 points are attributable to the dataset, 2.7 points to the parser and 7.9 points to the method. We further analyze error cases below.

4.3 Shortcomings of the data set

We found 49 errors imputable to shortcomings in the QA-SRL dataset in our sample. In 20 cases out of those, we found that the annotators chose an answer which is a semantic superset of the answer found in the passage. This situation is illustrated in Section 4.3.

(1) An error due to a superset relation between the gold and the retrieved answer

Passage: Placards on the courtyard wall explain it served as headquarters for Field marshal Kollowrat-Krakowsky battling Napoleonic forces in the 1796 Siege of Kehl

Question: Where was someone battling?

Gold: ‘Siege of Kehl’

Retrieved: in the 1796 Siege of Kehl

In this example, the correct answer is only “Kehl”, as the “siege of” indicates something which happened at “Kehl”. Thus, the gold provided by the annotators include the actual gold answer, but provide additional information.

Another issue that arises in the dataset (7 cases in our sample) is incorrect or incomprehensible questions. This is frequently caused by considering a word which is a noun or an adjective in the passage as verb (or part of a verb, e.g. a past participle in a passive verbal form) about which to ask questions. This concerns either homophonous forms or forms that can be formed by using a base form which is a homophone to the word in the passage. An example is shown below:

(2) An error due to changing the POS of a word in the passage

Passage: In 1977 a swamp created by heavy rains was found to contain 8 toxic materials, including 11 suspected cancer-causing chemicals

Question: When was something being swamped?

Gold: ‘in 1977’

In this example the noun ‘swamp’ is turned to a past participle, part of the passive past continuous verbal form “was being swamped”.

In the following example the incomprehensibility is caused by plain ungrammaticality:

(3) An error due to an ungrammatical question

Passage: A Texas man was rescued earlier this week after being adrift at sea for 31 hours, according to media reports on Monday

Question: Who was something according to?

Gold: ‘media reports’

Lastly, in 9 cases the actual answer is just not in the provided passage. Despite this problem, annotators did provide a gold answer. The following is such an example:

(4) An example where the answer is not in the passage

Passage: What this entails is a more complex relationship to technology than either techno-optimists or techno-pessimists tend to allow.

Question: What isn't being allowed?

Gold: 'complex relationship to technology', 'a more complex relationship to technology', more complex relationship to technology'

Here the passage tells us that "techno-optimists" allow do not allow *simple (or less complex) relationships to technology*. However neither the word "less" or "simple" or equivalent are found in the passage. Thus, the gold simply cannot be annotated as a span in the passage, even though annotators did attempt to do so.

Another notable issue is the incorrect identification of a verb occurrence which occurs more than once in the passage (the question is about one occurrence and the answer about another), accounting for two cases in our sample. In another two cases, the syntax of the passage was plainly incorrect, and thus the parser could not recover any useful UD tree.

4.4 Shortcomings of the parser

In most cases, parsing errors are attributable to difficulty in handling punctuation (in particular quotes)- and attachment errors.

In 5 out of the 13 parse error cases in our sample, Udify interpreted quotation marks as sentence final markers and terminated the parsing, as in the sentence: *After summarizing his career , Matisse refers to the possibilities the cut-out technique offers , insisting " [...] "* where the parser stops after the first quotation mark.

Another common error (6 cases out of 13) is incorrect attachment. That is, a subtree of the dependency tree is attached to the wrong head, as in: *Churchill was a prolific writer, often under the pen name " Winston S. Churchill " , which he **used** [...]* where "used" is attached to "writer" rather than "name". Of course, in this case, a correct attachment demands a fine understanding of the sentence, so one might wonder if this it reasonable to expect such precision from the parser. Indeed, this is precisely what we intend to estimate by our experiment.

4.5 Shortcomings of the method

Seen as a way to test parsers, our method relies on the assumption that predicate-argument relationships are either directly encoded in the UD syntax, or can be directly inferred from it. Thus, conversely, the predicate-argument relationship can serve as a proxy for testing UD parser. Even though the assumption generally holds (not withstanding parsing errors), it sometimes fails. In the rest of the section we analyze the cases when this happens.

Insufficient propagation of arguments The first class of issues is related to the propagation of argument to all the predicates where they apply. This sort of situation accounts roughly for one third of the errors attributable to shortcomings of the method. While EUD mandates subject control propagation, there are other kinds of argument propagation which can apply.

The first main case occurs when purpose clauses are present. Consider the following passage and question: "Public officials in Texas have urged citizens to receive a flu shot. Who receives something?" Here the answer can be retrieved from a relation between *citizens* and *receive*, but the relationship is not direct: it is mediated by a purpose clause, and this mediation is not identified explicitly in the UD representation.

The second main case involves topicalization of prepositional phrase. The following example illustrates. "In the summer, the glacier melts rapidly, producing a thick deposit of sediment. When is something produced?" In this case the temporal clause is not syntactically attached to *producing*. Rather, it is topicalized and thus attached to the top level node.

Semantic or pragmatic reasoning is necessary In the second class of issues, some sort of semantic and/or pragmatic reasoning is necessary to understand the relationship between arguments and their predicates. The following passage illustrates the problem: "New South Wales premier Mike Baird said people should leave work early and arrive home before dark, as storms were predicted to intensify. Why did someone say something?" Here the cause is not syntactically related to the verb "say". Furthermore, locating the cause cannot be a matter of traversing the syntax tree, using *any* method. Instead, proper identification of the answer relies on the lexical semantics of the passage. We attribute roughly one fourth of the shortcomings of the methods to this class. We stress however that the lines are blurred between various classes of errors. Even though the classification is done according to the best of our judgement it is not easy to make the difference between this case and the previous one.

Anaphora resolution Another cause of errors is the lack of anaphora resolution layer in the processing pipeline. For example, the search for syntactic arguments may find the pronoun "it", but the annotators could have resolved the anaphora to a noun phrase (say "the power plant"). This class of errors causes only a tenth of the method shortcomings. This low number may come as a surprise. Its relatively low weight can be explained by two factors: the first one is that annotators are allowed to point to pronouns when identifying arguments. In this case anaphora resolution plays no role. Additionally, each passage is only one sentence long. Therefore, the possibilities for anaphora resolution are limited.

Shortcomings of the parent heuristic When the answer is one of the children, we consider the whole subtree as a candidate answer. When the answer should be looked up in the parent node, we cannot do the same thing: the parent node would contain the whole phrase, which is wrong. For example, when trying to answer "Who observed?" given "The observed animals were tortoises", the parent is "animals", which is the root of the sentence. The heuristic that we apply is to subtract the subtree which contains the verb to obtain the candidate answer. Often, this works well, but in this example we obtain nonsense. This problem accounts roughly for 15 percent of method errors.

Other issues The above list covers roughly 80 percent of errors. The remaining issues include various idiosyncratic interpretations of passages and questions (parataxis, non-deterministic selection of non-specific relative clauses, etc.). Some of them seem as if they could be handled by special rules to identify arguments, but we have preferred not to implement such rules in order to keep the results more directly linked to the syntactic trees which we analyze.

5 Suggested improvements to annotation schemes

In this section we leverage our understanding of EUD and QA-SRL, and provide advice to creators of datasets featuring either annotation schema.

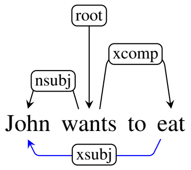
5.1 EUD

While the main UD format prescribes dependency *trees*, UD also specifies an enhanced format which allows for additional semantically relevant edges to be added (thus obtaining a graph). As Candito et al. (2017) among others note, different tasks seem to require different semantic representations. Thus, our suggestions to the EUD schema focus on how to extract arguments indicated by some question.

Our analysis shows that EUD is able to model the predicates and arguments in QA-SRL to a high degree (when probed with our fairly straightforward rule-based system) providing an appreciable increase in accuracy compared to plain UD, see Table 2. Yet, as far as we understand, the EUD annotation standard is lacking in clarity when it comes to how much semantic relations should be reflected in the structure. The standard reference appears to be the UD website[1], where all enhancements seem to be deducible algorithmically from the plain UD tree. However, as seen in Section 4.5, certain predicate-argument relationships are not present in the dependency structure, even after applying the algorithmic enhancements.

We believe that a variant of the EUD scheme with full reflection of predicate-argument structure would be beneficial for many downstream tasks. In the light of our experiment, we propose a number of following arcs to be added, as we list below.

EUD mandates the propagation of subjects through control verbs. As an illustration, consider the sentence "John wants to eat.". The UD tree contains the arcs in black, and EUD mandates to add the blue arc:

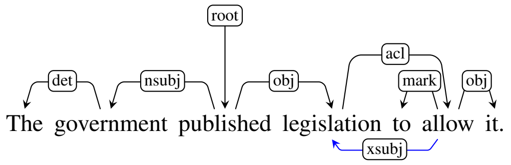


However, we have found that the predicate, the argument and the control verb are not arranged in fixed syntactic patterns, which makes adding the relevant arcs difficult. The main source of difficulties appear to be that the relationship between

[1] https://universaldependencies.org/u/overview/enhanced-syntax.html

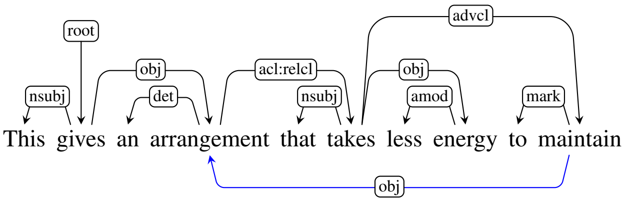
the argument and predicate can be mediated by a purpose clause.

To illustrate the complexity of the problem, we show two typical examples.



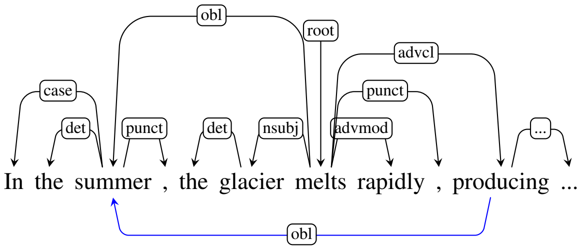
Above, the (semantic) subject of "allow" is "government", which is syntactically a grandfather node of allow. ("Legislation" is another candidate, but it also cannot be identified using a simple syntactic pattern.)

In the example below, we face two difficulties. First, "take" is not a control verb. Second, even though the desired argument of "maintain" (which is "arrangement") can be identified as an argument of "take", this can only be done *via* a relative clause. Third, the roles do not match (a subject becomes an object).


root
nsubj
obj
det
acl:relcl
nsubj
advcl
obj
amod
mark
obj
This gives an arrangement that takes less energy to maintain

In sum, we purport that, in general, the semantic subject (or object) of a predicate can be found anywhere in the sentence.

Another shortcoming observed is regarding topicalization. Topicalization occurs when a phrase in a sentence is moved to the front of the sentence, to make the phrase more prominent. In the case of prepositional phrases, often indicating semantic roles pertaining to the location, time, or manner in which something happens, is typically expressed with the role obl. However, two verbs may be associated with a prepositional phrase indicating time. Thus, the obl argument should be propagated similarly to how the subject and object roles are propagated in control-like verb construction. An example from the dataset, with out proposed enhancement in blue:



This addition allows for a straightforward interpretation of "when" things happen, by associating both "melts" and "producing" (which is a consequence of "melts") with the phrase "in the summer". This allows us to more easily extract the answer to the question: "when was something produced?".

Finally, anaphoric relationships should be noted as well. This is a well-studied topic which we won't comment further upon, however, we refer readers to the Universal Anaphora project (Poesio et al., 1999).

It should be noted that, contrary to the algorithmic transformations of UD trees, some of the above arcs cannot be deduced without a certain amount of semantic understanding of the sentence (in the sense that substituting lexemes by others with the same POS would change the structure). However, this kind of effect is already present when deciding the attachment of constituents, and therefore already affects plain UD.

5.2 QA-SRL

We have discovered several possible improvements regarding the QA-SRL data collection.

One prevalent source of ambiguity regards the selection of a general or specific phrase, as in Example (1). A way to remedy this ambiguity in future versions of the QA-SRL datasets is to give annotators more specific instructions for cases like these. A solution that seems to be viable is to instruct annotators to give the most specific answer as this is found in the text, which correctly answers the question. In plain words, this is the longest possible substring that correctly answers the question. In the case of Example (1), that would be the substring "in the 1796 Siege of Kehl". Note that the relations subset and superset have a more restricted meaning here, as they are bound by the specific syntax found in the passage. As such, the gold and the retrieved answer stand in a subset relation, if the former is a superstring (thus, more specific) of the latter and, vice versa, in a superset relation, if the former is a substring of the latter. An instruction to select the longer string would also lift the ambiguity inherent to selection of non-specific relative clauses. To illustrate, consider the passage-question pair "Matisse's wife Amélie , who suspected that he was having an affair, ended their 41-year marriage. Who ended something?" For this example the annotators marked 'Amélie' and 'Matisse's wife Amélie' as possible answers, but 'Matisse's wife Amélie , who suspected that he was having an affair' is the longest acceptable string.

To prevent incomprehensible questions (like Section 4.3), additional validation tests should be run to safeguard against the formation of ungrammatical questions. One way to do this is to validate at least part of the questions in the dataset using a syntactic acceptability task. This helps identify the ungrammatical questions and replace them with grammatical ones. We observed that annotators tend to make attempts at such meaningless questions as well as questions which do not have an answer in the passage. This is presumably caused by annotators "trying their best", but results in bogus answers. One idea to filter those would be to turn proposed answers into inference problems, as suggested by Demszky et al. (2018). If the constructed problem is not an entailment, then the answer should be rejected. For instance, Example 4 would be turned into the following problem:

(5) NLI pair for Example (4)

Premise: What this entails is a more complex relationship to technology than either techno-optimists or techno-pessimists tend to allow.

Hypothesis: Complex relationship to technology isn't being allowed.

Even though the double-negation complicates reasoning, in this case, one can reasonably expect that the absence of entailment could be detected. This could be done by another round of annotations, perhaps helped by a statistical model which would select doubtful cases.

6 Related Work

In addition to our suggestions, there has been several other proposals to extend syntactic dependency trees to more explicitly cover semantic phenomena, including the work of Silveira (2016), already discussed in the introduction.

Additionally, Candito et al. (2017) notably propose additions to the EUD schema mainly focusing on extracting the arguments of non-finite verbs and dealing with syntactic alterations in a French treebank.

The Universal Decompositional Semantics project (White et al., 2016; Zhang et al., 2017) is another attempt at extending the UD framework to cover semantic phenomena. They develop the Semantic proto-role labeling protocol (SPR1 and SPR2), to find proto-semantic roles by decomposing semantic roles such as "Agent" into more fine-grained properties.

Working more generally on dependency trees, Stanovsky et al. (2016) develop a framework to enhance dependency trees such that semantic propositions are more easily recoverable which includes a similar propagation of subjects and objects as in EUD. However they do not appear to take any special note of purpose clauses or topicalization.

7 Conclusion and Future Work

We have found that a state-of-the-art UD parser such as Udify only fails to produce a semantically correct UD trees in rare cases. If we exclude difficulties in handling quotes, only 8 cases out of 100 errors are imputable to the parser.

However, in a lot of cases the semantic relationship cannot possibly be present in the UD format, due to its tree structure. To express this, enhancing the structure with additional arcs is needed. Some of those arcs can be found by algorithmic means (as listed in Section 3), boosting the accuracy by a several points, see Table 2. One could expect that the EUD schema would mandate the addition of all semantically relevant arcs, but this is not the case. We have advocated for an update to the EUD standard which fills this gap, as discussed in Section 5.1.

While the goals of the QA-SRL appear to align perfectly with ours, and the annotation for QA-SRL was both effective and relatively cheap, we notice some shortcomings in the annotations (Section 4.3). Sometimes annotators get something wrong because of a tricky phenomena or they are presented with a badly formulated question about the passage. We have proposed a number of strategies to improve data collection for future similar datasets (Section 5.2). Another point to consider is that it is much cheaper to annotate QA-SRL than full EUD parse trees. Therefore QA-SRL could be a proxy for training EUD parsers on predicate-argument structures, together with for example multi-task learning. That is, in addition to training a system to predicting arcs, the system would be optimized on selecting the spans of text corresponding to the arguments of predicates.

Acknowledgements

We would like to thank the reviewers for their helpful comments. The research reported in this paper was supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Gosse Bouma, Djamé Seddah, and Daniel Zeman. 2020. Overview of the iwpt 2020 shared task on parsing into enhanced universal dependencies. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 151–161.

Gosse Bouma, Djamé Seddah, and Daniel Zeman. 2021. From raw text to enhanced universal dependencies: The parsing shared task at iwpt 2021. In *Proceedings of the 17th International Conference on Parsing Technologies (IWPT 2021)*, pages 146–157.

Ekaterina Buyko and Udo Hahn. 2010. Evaluating the impact of alternative dependency graph encodings on solving event extraction tasks. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 982–992.

Marie Candito, Bruno Guillaume, Guy Perrier, and Djamé Seddah. 2017. Enhanced UD dependencies with neutralized diathesis alternation. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 42–53, Pisa,Italy. Linköping University Electronic Press.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *CoRR*, abs/1809.02922.

Adam Ek and Jean-Philippe Bernardy. 2020. How much of enhanced UD is contained in UD? In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 221–226, Online. Association for Computational Linguistics.

Murhaf Fares, Stephan Oepen, Lilja Øvrelid, Jari Björne, and Richard Johansson. 2018. The 2018 shared task on extrinsic parser evaluation: On the downstream utility of English Universal Dependency parsers. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 22–33, Brussels, Belgium. Association for Computational Linguistics.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. 2018. Large-scale QA-SRL parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2060, Melbourne, Australia. Association for Computational Linguistics.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Anshuman Mishra, Dhruvesh Patel, Aparna Vijayakumar, Xiang Li, Pavan Kapanipathi, and Kartik Talamadupula. 2020. Reading comprehension as natural language inference:a semantic analysis. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 12–19, Barcelona, Spain (Online). Association for Computational Linguistics.

Makoto Miwa, Sampo Pyysalo, Tadayoshi Hara, and Jun'ichi Tsujii. 2010a. Evaluating dependency representations for event extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 779–787, Beijing, China. Coling 2010 Organizing Committee.

Makoto Miwa, Sampo Pyysalo, Tadayoshi Hara, and Jun'ichi Tsujii. 2010b. A comparative study of syntactic parsers for event extraction. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 37–45.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Timothy Osborne and Kim Gerdes. 2019. The status of function words in dependency grammar: A critique of universal dependencies (ud). *Glossa (Online)*.

Massimo Poesio, Florence Bruneseaux, and Laurent Romary. 1999. The mate meta-scheme for coreference in dialogues in multiple languages. In *ACL'99 Workshop Towards Standards and Tools for Discourse Tagging*, pages 65–74.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 89–101, Copenhagen, Denmark. Association for Computational Linguistics.

Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D. Manning. 2015. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the Fourth Workshop on Vision and Language*, pages 70–80, Lisbon, Portugal. Association for Computational Linguistics.

Sebastian Schuster and Christopher D Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2371–2378.

Natalia G Silveira. 2016. *Designing syntactic representations for NLP: An empirical investigation*. Ph.D. thesis, Stanford University.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. 2016. Getting more out of syntax with props. *arXiv preprint arXiv:1603.01648*.

Mingming Sun, Wenyue Hua, Zoey Liu, Xin Wang, Kangjie Zheng, and Ping Li. 2020. A predicate-function-argument annotation of natural language for open-domain information expression. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2140–2150.

Aryeh Tiktinsky, Yoav Goldberg, and Reut Tsarfaty. 2020. pybart: Evidence-based syntactic transformations for ie. *arXiv preprint arXiv:2005.01306*.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723.

Sheng Zhang, Rachel Rudinger, and Benjamin Van Durme. 2017. An evaluation of predpatt and open ie via stage 1 semantic role labeling. In *IWCS 2017—12th International Conference on Computational Semantics—Short papers*.