Rule-based semantic interpretation for Universal Dependencies

Jamie Y. Findlay and **Saeedeh Salimifar** and **Ahmet Yıldırım** and **Dag T. T. Haug**
Department of Linguistics and Nordic Studies
University of Oslo

Abstract

In this paper, we present a system for generating semantic representations from Universal Dependencies syntactic parses. The foundation of our pipeline is a rule-based interpretation system, designed to be as universal as possible, which produces the correct semantic structure; the content of this structure can then be filled in by additional (sometimes language-specific) post-processing. The rules which generate semantic resources rely as far as possible on the UD parse alone, so that they can apply to any language for which such a parse can be given (a much larger number than the number of languages for which detailed semantically annotated corpora are available). We discuss our general approach, and highlight areas where the UD annotation scheme makes semantic interpretation less straightforward. We compare our results with the Parallel Meaning Bank, and show that when it comes to modelling semantic structure, our approach shows potential, but also discuss some areas for expansion.

1 Introduction

Quite aside from the deep theoretical interest in discovering how syntactic information contributes to semantic interpretation, there are also a number of practical benefits to augmenting syntactic descriptions with semantic representations. A suitably rich semantic representation automatically makes possible a number of common downstream tasks such as named entity recognition, information retrieval, machine translation, and natural language inference. In this paper, we report on our system for using Universal Dependencies syntactic annotations (UD: Nivre, 2016) to produce semantic representations, in this case Discourse Representation Structures (DRSs: Kamp and Reyle, 1993; Kamp et al., 2011). Figure 1 shows the UD parse and a possible DRS representation for a simple sentence.

In particular, and unlike much of the state of the art, our pipeline makes heavy use of a rule-based

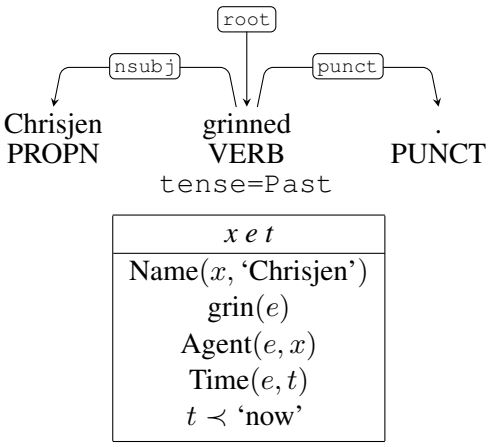

Figure 1: UD graph and DRS for *Chrisjen grinned*

component. This component inspects the UD graph and uses it to produce a number of *meaning constructors*, the basic building blocks of semantic composition in Glue Semantics (Glue: Dalrymple et al., 1993; Asudeh, 2022). Meaning constructors are pairs, the first element of which is a lambda expression in some meaning language, and the second element of which is a formula in linear logic (Girard, 1987) that expresses a type. The atoms of this linear logic statement are indexed with node labels, thereby anchoring (or 'gluing') the semantics to the syntax. This flexible approach to meaning composition allows each word to make any number of distinct meaning contributions, and frees composition from word order, making it a perfect fit for a dependency grammar like UD (see Haug and Findlay 2023).

Rules in our rule-based system consist of two parts: on the left-hand side, a description which nodes in the UD tree might satisfy (e.g. referring to the node's dependency relation, its lemma, or its features), and on the right-hand side, a meaning constructor to be introduced. This system has been implemented, using a Haskell script to inspect the UD tree node by node, comparing each one to the rules in our ruleset, and introducing the appropriate

meaning constructor each time a node matches a description (for more details on this process, albeit in a different syntactic setting, see Gotham and Haug 2018).[1]

Once a collection of meaning constructors has been obtained, they are passed to the Glue Semantics Workbench (Messmer and Zymla, 2018), which uses them to produce a linear logic proof (or proofs, in the case of scope ambiguities) whose conclusion is the meaning constructor corresponding to the semantic representation of the sentence. We subequently use the Python Natural Language Toolkit (NLTK: Garrette and Klein, 2009) to perform any post-processing steps, including producing human- and machine-readable DRS outputs.

Our system is part of an ongoing project on universal semantic parsing, and so another prominent feature of our system is its focus on broad coverage. This sets it apart from other works which combine symbolic and machine-learning approaches (e.g. Kalouli and Crouch 2018; Hu et al. 2020), since these are limited to specific languages, e.g. English, because specific tools exist, or to other languages for which there exist sufficient data to train a deep learning system. Because of the lack of semantically annotated training data for the majority of the world's languages, recent efforts in broad coverage semantic parsing (e.g. Liu et al. 2021) have been based on machine translation into English, followed by semantic parsing and projection of the result onto the source language. However, state of the art machine translation is only available for high-resource languages (Haddow et al., 2022) and is likely to introduce noise even in the best of cases, especially if the languages are typologically distant.

Instead of this translational approach, we try to leverage UD representations to achieve universality. As far as possible, our rule system produces meanings based exclusively on the UD parse, without invoking language- or lemma-specific rules. Section 2 discusses the kinds of rules used in more detail (and Section 3 identifies some issues that arise which are of potential relevance to UD as a framework). However, this language-neutral approach means that the output of the rule-based component is necessarily underspecified, since, for example, semantic roles (Agent, Patient, etc.) do not stand in a one-to-one correspondence with syntactic relations (`nsubj`, `obj`, etc.). For some languages, this is as far as we can go. But where languages have more resources available, and we can therefore access the language- and lemma-specific information needed, we can make use of various post-processing steps to further refine our semantic representations. One of these systems, used to convert syntactic labels for dependencies into appropriate semantic role labels, is described in Section 4.

In Section 5, we compare the output of our pipeline with an existing benchmark of DRS parsing, the Parallel Meaning Bank (PMB: Abzianidze et al., 2017). Our goals are slightly different from those of the PMB, so although this comparison offers indications about the adequacy of our rule system, it does not offer a perfect gold standard.

Moving forward, we have further plans for post-processing, and these are discussed in Section 6. We also indicate some limitations of the PMB dataset as a gold standard for DRS parsing.

2 Rules for semantic interpretation

By using a rule-based system, we can more easily import insights from theoretical linguistics into automatic semantic interpretation. These insights are generally of a structural nature: e.g. the fact that the logical structures produced by different quantifiers do not straightforwardly match their syntactic structure is the sort of thing that may be difficult for a machine-learning algorithm to infer.[2] At the same time, our universal goals mean that language-specific information, such as the semantic roles a predicate assigns to its arguments, must be abstracted away from, since we cannot retrieve this information from the UD parse alone. The target output of our rule-based system is therefore not a fully-specified DRS. Instead, we aim to produce a *structurally* accurate DRS, where the correct discourse referents are present and the hierarchical relations between them are correct; the *content* of the DRS, by which we mean the labels for the relations, or the word senses attributed to the discourse referents, will be filled in only later, by language-specific post-processing. Concretely, except for in the cases where no language-/lemma-specific information is required to determine the correct labels, our rule system outputs syntactic (rather than semantic) labels for the relations between discourse

[1] The code used for our system is available at `https://github.com/Universal-NLU/UNLU`, including a sample set of semantic interpretation rules.

[2] By contrast, tasks like word sense disambiguation, which rely on large numbers of sometimes subtle cues, are precisely those tasks for which machine-learning systems are well suited.

referents, and uses lemmas in place of word senses.

2.1 Target representations

As mentioned, our target semantic representations are DRSs. In order to facilitate comparison with an existing benchmark, we aim to follow the specific format of the Parallel Meaning Bank (PMB). This is a fairly standard meaning representation format based on a neo-Davidsonian event semantics whereby verbs denote predicates of events (or states) and participants in these events are connected via thematic role predicates like Agent and Patient that relate events and individuals (Davidson, 1967; Parsons, 1990). The PMB does make a few less standard choices, however. For example, it is less expressive than some semantic theories in that it has no representation of number (except for in the case of 1st and 2nd person pronouns); but it is also more expressive in that it annotates a basic level of presuppositional structure (based on Projective DRT: Venhuizen et al. 2013). Ultimately, we wish to improve on both of these areas, by incorporating an explicit representation of number, and by capturing more presupposition triggers, but for now we attempt to diverge as little as possible from the PMB representations, in order to facilitate comparison.

2.2 Types of rule

In this section, we illustrate a few categories of rule, divided by the kind of information they require from the UD parse.

2.2.1 Part of speech

For some situations, the part of speech tag alone is sufficient to determine the node's semantic contribution. This is the case for proper nouns, for example, since we know they will contribute a discourse referent that stands in the 'Name' relation to its lemma (its name). Our rule that captures this is shown as rule 1 in Figure 2 (we also employ a second rule, not shown, that provides a meaning constructor that turns this meaning into a generalised quantifier). If a node has the UD POS `PROPN`, then we introduce a meaning constructor of type `e(!) -o t(!)` that adds the appropriate condition to the DRS for the sentence. The semantic side of the meaning constructor is written in the DRS representation language of the NLTK. On the linear logic side, we use `!` and `^` to refer to the current node and its mother, respectively; these will be instantiated to numeric node indices in a specific parse. The string `-o` is used to represent the linear implication symbol $\multimap$; Glue Semantics uses linear logic to guide semantic composition, following the 'proofs-as-programs' paradigm enabled by the Curry-Howard isomorphism (Curry and Feys, 1958; Howard, 1980). So the linear logic expression in this rule tells us it has the type $\langle e, t\rangle$ and that it is anchored in the current node, the one with the POS `PROPN`.

2.2.2 UD tree

For other cases, the topography of the UD tree itself encodes the semantic information we wish to capture. For example, some syntactic dependencies are also semantic dependencies – arguments and adjuncts like `nsubj`, `obj`, `ccomp`, `obl`, etc. We therefore require a number of rules whereby the presence of such a dependency produces a meaning constructor that introduces a parallel semantic dependency. Rule 2 in Figure 2 shows an example for `nsubj` when it is a dependent of a verb. This rule has two conditions, joined by `;`, signifying conjunction: the UD dependency of the node must be `nsubj`, and its mother node must have the POS `VERB`. We employ a Champollion-style representation of verbal meanings such that they do not have the usual $\langle v, t\rangle$ type of properties of events, but rather the higher type $\langle\langle v, t\rangle, t\rangle$ (Champollion, 2015). To minimise clutter in our rules, we define a new type `x(n)` which is equivalent to `((v(n) -o t(n)) -o t(n))`. The meaning constructor in rule 2 therefore consumes a generalised quantifier and produces a modifier of verbs, which adds the verb to the scope of the quantifier, and connects the variable being quantified over to the verb's event variable via an `nsubj` relation.

Although in general we require language-specific valency lexica to know which semantic role labels to use in place of syntactic labels like `nsubj`, in some cases we can nonetheless incorporate word-level information to make our DRSs more informative. For example, for `obl` dependents which have a `case` daughter, we use the lemma of the target of `case` (i.e. the preposition name) to label the semantic relation, thus adding a degree of granularity which would otherwise be absent.

Not all syntactic dependencies also correspond to semantic dependencies, of course: more functional ones like `aux`, `cop`, `case`, etc. usually do not in themselves (i.e. merely by their presence) contribute semantic information that is not also

```
1. coarsePos = PROPN -> \X.(([], [Name(X, ':LEMMA:')])) : e(!) -o t(!)
2. relation = nsubj; ^ {coarsePos = VERB} ->
      \Q.\V.\F.(Q(\X.(V(\E.(([], [nsubj(E,X)]) + F(E)))))) :
      ((e(!) -o t(^)) -o t(^)) -o (x(^) -o x(^))
3. coarsePos = VERB; ~ aux; Tense = Pres ->
      \V.\F.(V(\E.(([T], [time(T), EQ(T, 'now'), Time(E, T)]) + F(E)))) : x(!) -o x(!)
```

Figure 2: Some semantic interpretation rules

represented elsewhere; rather, the targets of such dependencies contribute semantic information in other ways, such as via their features.

2.2.3 Features

The UD feature space is not as consistently or reliably employed in treebanks as the part of speech tags or dependency graph labels are, and so we use it only sparingly in our rule system. Nonetheless, there are certain cases where it supplies crucial information that saves us having to fall back on language-specific resources. For example, the tense of simplex verbs (those without auxiliaries) can be reliably read off the `Tense` feature, as rule 3 in Figure 2 illustrates for the present tense (the symbol ~ represents negation).

2.2.4 More complex constructions

Of course, such simple rules only get us so far. Other phenomena, such as coordination or negation, require a rich set of complex interacting rules. Coordination is made especially challenging by the fact that in UD there is no node which represents the coordinate structure as a whole, meaning that the line between properties of the whole structure and properties of the first conjunct is blurred. There are other complexities here too: for instance, our system currently assumes that coordination is only possible with identical UD relations (e.g. coordinated `objs`), since the relationship between each conjunct and its semantic governor is mediated through the first conjunct, so whatever UD relation that word bears is assumed to be extended to the rest of the conjuncts. But of course this is empirically inadequate: as Przepiórkowski and Patejuk (2018) point out, in a sentence like *He asked her for a kiss and to go on a date with him* the first conjunct is an `obl` but the second would be annotated as an `xcomp` if it stood alone. Yet here it is merely a `conj` daughter of *kiss*, so it is not easy to reconstruct a different relationship with the verb than the one it bears indirectly via its mother. To some extent we can leverage the fact that UD relations are partly determined by the part of speech of the dependent: for example, a verbal `conj` dependent of a noun will be a `csubj` if the noun is `nsubj`. But if the noun is `obj`, it can be `ccomp` or `xcomp` and we won't always have the morphological features to decide, especially not in a universal setting. Finally, if the noun is `obl`, as in the example from Przepiórkowski and Patejuk (2018), we run into the problem that UD makes an argument/adjunct distinction for clauses but not for noun phrases: if the sentence was *He did it for the money and to please his record company*, the infinitive would be `advcl`.

2.3 Challenges of universalism

To a large extent, our more targetted goal of obtaining the correct semantic *structure* while abstracting away from specific labels means that we do not rely on language-specific information, and therefore can develop a genuinely universal rule scheme which relies solely on properties of the UD parse. However, there are certain aspects of semantic structure where language-specific information may still be required. For example, the semantic structures arising from universal vs. existential quantifiers are different, and nothing in the UD parse encodes this distinction. We therefore maintain a small list of parameters whose values are language-specific lemmas which identify certain key words, such as the form of universal and existential quantifiers. We also currently encode the form of future auxiliaries (e.g. English *will*), infinitive markers (e.g. Eng. *to*), and definite determiners (since we find that the use of features like `Definite=Def` or `PronType=Dem` in treebanks and parsers is inconsistent). When parsing a language which lacks this data, we default to more coarse-grained rules which rely more heavily on features, or simply collapse some distinctions.

Similarly, there can be high-level grammatical differences between languages, such as whether they employ ergative or accusative case-marking, or whether they make use of negative concord, which are also relevant to the task of building a semantic structure. To capture these, we parametrise

some rules, so that we can specify for each language which version should apply. When handling a language for which we lack this information, we assume the most typologically common version of the rule.

There are also very low-level lexically determined properties of semantic structure, e.g. the well-known distinction between subject-control and object-control verbs like *promise* and *persuade*: the UD trees for sentences like *I promised Holden to come* and *I persuaded Holden to come* will be identical, but the semantic argument of the subordinate clause's verb is different in each case (*I* in the first, *Holden* in the second). Given our goals, such information will unfortunately be missed; the requirement that a UD parse produce a tree (as opposed to a more general kind of graph) means that the syntactic representation we start from is not as rich as it would be in other frameworks (since there is no re-entrancy, for example), and certain information is therefore simply not represented.

3 Implications for Universal Dependencies

3.1 Shallowness of representation

This relative shallowness of UD parses is a well-known shortcoming of the framework. Despite the putative advantages of working with more constrained trees rather than full-fledged graphs, we wish to add our voices to those who believe the costs of this limitation outweigh the benefits. If UD annotations included the controllers of `xcomps`, for instance, then the problem mentioned above would not arise, since the difference between *promise* and *persuade* would also be indicated syntactically. This is done in so-called Enhanced UD (cf. Schuster and Manning 2016), for instance, but the cross-linguistic coverage of Enhanced UD treebanks is currently *much* sparser than basic UD (contentful Enhanced UD annotations are only available for 31 out of the 213 UD treebanks, and of these only 22 contain annotations indicating the controllers of `xcomps`). Although there exist automatic 'enhancers' which can convert basic UD into Enhanced UD (e.g. Nyblom et al. 2013; Schuster and Manning 2016; Nivre et al. 2018; Bouma et al. 2020), they are either language-specific or quite rudimentary (see critique in Findlay and Haug 2021). While it would certainly be useful to produce more Enhanced UD treebanks, we do not think it is likely that this will happen on the same scale as the UD project generally, and it is especially unlikely for low-resource languages, so we continue to make use of the basic UD annotations in our universally-oriented project.

3.2 Pro-drop

The problem of missing controller annotations intersects with another problem, discussed by Patejuk and Przepiórkowski (2018) – that of unexpressed/pro-dropped arguments. Since pro-dropped arguments are not present in the string, they are not included in a UD parse, and this makes semantic interpretation much more challenging. We must always allow for the possibility that there are additional discourse referents which are related to each predicate in an unspecified way; and without accessing language-specific valency information we have no way of knowing how many or what kind of dependents might have been omitted. This issue also means that control relations cannot be included even in Enhanced UD representations when the controller is itself an unexpressed argument. We therefore agree with Patejuk and Przepiórkowski (2018, 216ff.) and Przepiórkowski and Patejuk (2020, 205ff.) that the inclusion of empty nodes in the string to represent pro-dropped arguments would be a valuable addition to basic UD (and would also help with adding control annotations: see Findlay and Haug, 2021, 26f.)

3.3 Lexical focus of features

UD feature annotations are scrupulously limited to the word level. This is problematic when features of phrases emerge non-compositionally, e.g. in periphrasis. As the UD guidelines acknowledge, "If a tense is constructed periphrastically [...] and none of the participating words are specific to this tense, then the features will probably not directly reveal the tense".[3] In this view of things, features like `Tense` should be seen as morphological features: they describe nothing more than the form of individual words, which may happen to align with their syntactic properties, but if so then only incidentally. However, such a view is at odds with the guidelines' own definition of the `Tense` feature: "Tense is a feature that specifies the time when the action took/takes/will take place, in relation to a reference point". This is an emphatically semantic definition. But given the problem of periphrasis, the `Tense`

[3] https://universaldependencies.org/u/feat/Tense.html

feature cannot be given any definitive semantic interpretation; the presence of `Tense=Pres` in English, for example, does not guarantee any reference to present tense – one of the places it occurs is on *-ing* participles, even when used in the past, as in *They were singing*. And `Tense=Past` appears on passive participles in English (whatever the tense), since they share the same form as past participles (further evidence this is in fact a morphological feature). While it would be possible to write rules to translate each language's unique combinations of morphological forms into the correct tense interpretations, this clearly goes against the universal aims of our project, and of UD itself. We believe therefore that it would be advantageous for UD to adopt some notion of clause-level features for qualities such as tense which are not usefully localised at the word level, or to concede that such features are purely morphological, and do not encode the semantic information they are currently claimed to.

4 Post-processing

A full semantic representation contains many types of information that simply cannot be extracted from the UD tree, even with the aid of linguistically-informed rules. Typically, this is information that would be associated with lexical entries rather than with structural syntax.

The most prominent example of this kind is the mapping from syntactic functions to semantic roles: UD gives us labels like `nsubj`, `obj` etc., but how these map to roles like Agent, Patient, Stimulus, Experiencer, etc. is verb-specific. We resolve this mapping in a separate post-processing step, where for English we rely on VerbNet (Kipper et al., 2008), which provides details of the syntactic frames of English verbs and their associated semantic roles.

VerbNet arguments are specified in terms of syntactic categories with associated selectional restrictions, which we translate into regular expressions over relations resulting from our UD translations – basically syntactic roles or prepositions. Figure 3 shows our translations of some of the frames that VerbNet version 3.3 specifies for the verb *look*.

To choose the VerbNet frame to use, we pick the frame that has the fewest items not present in the DRS; if there is a tie, we reject all frames that do not specify core relations (`nsubj`, `csubj`, `obj`, `iobj`, `xcomp`, `ccomp`) that are present in the DRS, and pick the remaining one that has the fewest relations in the DRS not present in the frame; and if it still not unique, we keep both options. Notice that we minimize elements in the frame not present in the DRS before the opposite, because the DRS will in many cases contain adjunct relations that are not specified by VerbNet frames.

As an example, consider the sentence *How do people look at and experience art?*. In our translated DRS, the looking event bears three relations: an `nsubj` relation (to the discourse referent of *people*), an `at` relation (to the discourse referent introduced by *art*), and a *how* relation (to some discourse referent (a state) whose identity is asked for). Of the frames in Figure 3, we choose the second one, because it specifies two elements that are both present in the DRS, whereas the other two frames contain elements that are not in the DRS. None of the frames tell us anything about the *how* relation, which should ideally be spelled out as Manner, so this must be resolved in a different way.

5 Comparison with the PMB

In order to assess how well our rule-based system performs, we conducted some experiments comparing our outputs to the German, English, Italian, and Dutch gold standard datasets (produced and checked by human annotators) provided by the Parallel Meaning Bank v. 4.0.0 (Abzianidze et al., 2017). We compare the pipeline output with the test sets of these languages using the Counter tool (van Noord, 2022), which enables the comparison of two DRSs that are expressed in a machine-friendly format called 'clause notation' (see Liu et al. 2021 for details about this notation). We use the automatic parser Stanza (version 1.4.0: Qi et al., 2020) to produce the Universal Dependencies representations which serve as input to our pipeline.

In the clause notation, lexical concepts are referred to via their WordNet synset (Fellbaum, 1998); e.g. the concept expressed by the lemma *man* might be represented as 'man "n.01"'. At present, our pipeline does not deal in this level of lexical granularity, instead simply outputting lemmas as DRS conditions. For the purpose of comparison, we therefore assign all lexical concepts a default WordNet sense, suffixing all such conditions with "n.01".

Our first consideration is coverage. There are a number of cases where the pipeline fails to produce a DRS for a given sentence, and therefore comparison with the PMB would be unilluminating. There are three main causes:

‘(csubjlnsubj)’: ‘Agent’, ‘over’: ‘Location’, ‘through’: ‘Location’, ‘into’: ‘Location’
‘(csubjlnsubj)’: ‘Agent’, ‘PREP’: ‘Theme’
‘(csubjlnsubj)’: ‘Agent’, ‘PREP’: ‘Location’, ‘for’: ‘Theme’

Figure 3: Select translated VerbNet frames for *look*

1. Faulty input: sometimes Stanza fails to produce a sensible input for the pipeline. For example, Stanza sometimes incorrectly interprets ‘.’ in German ordinal number expressions as the end of a sentence, and therefore produces inappropriate and often ungrammatical parses.

2. Computation takes too long for a sentence: in case running the linear logic proof takes too much time, we automatically stop the computation after 10 minutes for that sentence. This could merely be a question of optimisation, or might point to issues with certain interactions of our rules.

3. Genuine lack of coverage: our system is still a work in progress, and there are several linguistic phenomena which we do not even attempt to cover at present. One large omission is negation, for instance. Sometimes these gaps merely lead to inaccurate DRSs, but sometimes they make it impossible to derive a complete DRS at all. Although this points to areas where more work is required, failure in these cases does not tell us anything about the accuracy or usefulness of what we *have* implemented.

For this reason, we omit from our comparisons those sentences where we fail to produce a DRS. Coverage ranges from 79–93% – see Table 1.

Table 1 also shows the results of comparison between the output of our pipeline and the PMB gold data. Where we compare our output directly with the gold data (the ‘raw’ comparison condition), two things are clear: the scores for English are much better than for the other languages, and all four sets of scores are not particularly impressive. This is shown more perspicuously in Figure 4. Why should this be the case?

There are two main reasons for the discrepancy between the English and non-English scores. Firstly, the PMB uses English synsets for all languages, whereas our pipeline uses lemmas for the equivalent conditions, and these are not translated

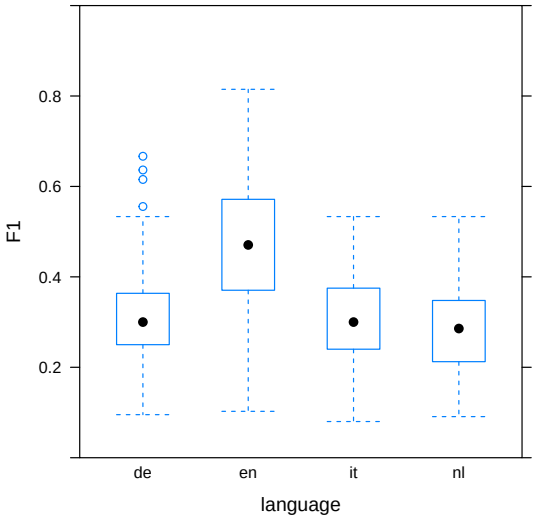

Figure 4: Raw F1 scores across languages

(e.g. where an Italian text uses *uomo*, we will produce a DRS condition ‘uomo "n.01"’, whereas the PMB gold will have ‘man "n.01"’). This means that we will systematically score worse in non-English languages, since almost every single condition which comes from a lexical concept will differ from the PMB gold, and so be scored down in comparison. Secondly, we have only implemented the semantic role labeling step described in Section 4 for English, so once again the non-English languages contain a number of systematic errors: all relations between discourse referents will be wrong since they will have syntactic rather than semantic labels.

The English scores, though, are still not particularly impressive. However, there are a number of things being compared here which we make no effort to cover, and so are bound to do badly on. For instance, we make no effort to find the correct synset for lexical concepts.

Since our focus is on obtaining the correct semantic *structure*, a more illuminating comparison would be to compare the structures of our DRSs, ignoring specific role or concept labels. This was achieved using Counter’s `-dr` (default role), `-dc` (default concept), and `-dse` (default word sense)

Language	Raw comparison F1	Rec	Prec	Structural only F1	Rec	Prec	Covered sentences	Total sentences	Proportion covered
German (de)	30.78	30.85	31.21	59.58	57.39	63.48	434	547	0.79
English (en)	46.69	48.07	46.28	63.42	63.92	64.32	874	1048	0.83
Italian (it)	30.68	30.55	31.40	58.88	57.25	61.92	429	461	0.93
Dutch (nl)	28.63	29.07	28.84	58.41	56.84	61.59	399	491	0.81

Table 1: Average F1, Recall, and Precision percentage scores for the sentences covered by our pipeline in the raw and structural-only comparison conditions, followed by number of sentences receiving an analysis, total number of sentences in the data set, and the corresponding proportion of sentences covered

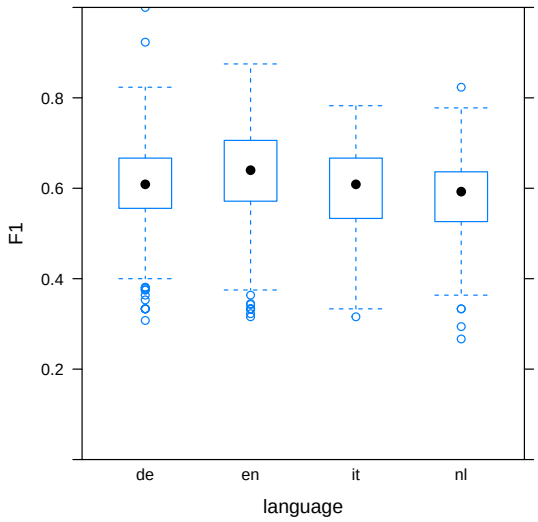

Figure 5: Structural comparison F1 scores across languages

flags, which overall ignores the effect of getting role, concept, or word sense/synset labels incorrect. This enables us to compare DRSs in purely structural terms (along with non-language-specific, discourse-related conditions like PRESUPPOSITION), without worrying about the content of the relations or lexical concepts introduced. F1 scores for this comparison are shown in the second part of Table 1 and visually in Figure 5.

In this setting, the stark difference between English and the other languages disappears, and the scores improve markedly. Some of the higher individual sentence scores are very good, but the averages are dragged down by some very poor scores as well. We anticipate that as the coverage of the rules is expanded, the number of such poorly scoring sentences will diminish, and the overall scores will correspondingly improve.

Our pipeline still performs slightly better on English even in this structural-only setting, which is likely due to the fact that we have so far used English as our primary test language during development of the rules. On the positive side, the fact that performance is quite even across the other three languages, and not much lower than English, shows that our system generalises nicely outside of English. However, it would of course be nice to have gold data from less typologically similar languages to test this further.

Previous work on DRS parsing with neural methods has reported F1 scores in the high 80s on the PMB data (see van Noord et al. 2020). The results of our rule-based pipeline may seem abysmal in comparison, therefore. However, further testing has shown that the rule-based system degrades far less as sentence length increases, and may therefore be more robust. Most sentences in the PMB test set are very short: the vast majority are shorter than ten tokens, and the average length is 6.7. To test performance on longer sentences, we annotated Wikipedia text from the GUM corpus (Zeldes, 2017) with the PMB tool. The average sentence length in this data set is 19.5 tokens. Taking into consideration only data for which a non-zero F1 score is obtained (around 80% of the data for the DL models, and around 60% for our pipeline), Figure 6 shows the F1 scores for our pipeline with gold UD (ud), our pipeline with automatically-generated UD (stanza), a neural parser with no pre-training (no-pt), and a neural parser with the pre-trained bert_base_cased (bert_cased) language model.[4] The neural approaches suffer a major drop in performance compared with the PMB data, while our system suffers a less pronounced degradation. We believe this gives us reason to believe that as our

[4]Here, the no-pre-training and pre-trained models are sequence-to-sequence (seq2seq) models based on common practices for this type of task (cf. Zoph et al., 2016; van Noord et al., 2020; Gheini et al., 2021). The encoder side of the seq2seq model is either a no-pre-training model to be trained or a pre-trained (frozen) model such as bert_base_cased.

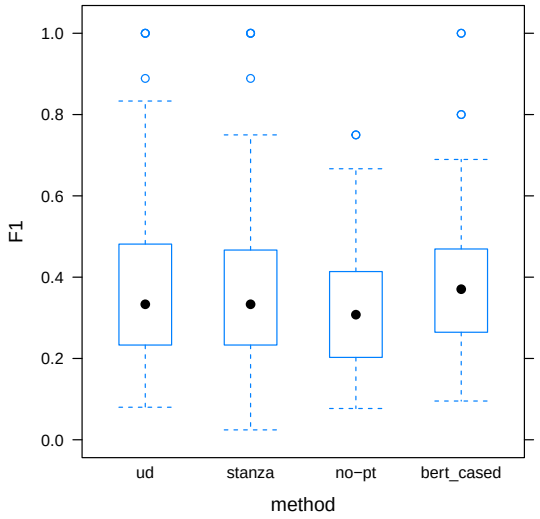

Figure 6: Performance on covered data in GUM corpus

system's rule coverage is improved, performance on both datasets will improve commensurably.

6 Going beyond the PMB

Along with improving our rule coverage to bring us more closely in line with the PMB, there are other areas where we intend to go beyond the PMB and capture additional phenomena via further post-processing steps.

Presupposition is one such area: at present, the DRSs included in the PMB are in the format of Projective Discourse Representation Theory (PDRT: Venhuizen et al., 2013, 2018), and so in principle have access to a rich set of tools for handling the similarities and differences between the information status of various types of projective content such as presupposition, anaphora, and conventional implicature. Setting anaphora to one side for the moment, the projective content the PMB currently contains consists of presuppositions triggered by proper names, definite descriptions, pronouns, and possessives (Abzianidze et al., 2019). Our pipeline also currently captures these, since their triggering contexts are legible in the UD parse (assuming suitable part of speech tags and lexical features). We are currently conducting experimental work to determine other presupposition triggers which can be incorporated into our pipeline, some perhaps on a language-specific basis. The lack of more presuppositional content in the PMB means it does not at present live up to the potential afforded by its PDRT underpinnings.

Anaphora resolution is an essential step in semantic interpretation, which not only changes the labelling of a DRS, but also affects its structure. We do not currently implement any anaphora resolution, whereas the PMB does, so it may be that this is another area where our scores have been dragged down. However, since the PMB dataset consists of fairly short sentences, there will be fewer opportunities for this to make a significant difference. What is more, the PMB's anaphora resolution is apparently fairly unsophisticated, and linguistically naïve: for example, it violates well-established constraints on binding, as in this Principle B violation from the English test data: *Tom$_i$ never spoke of him$_i$*. A more robust anaphora resolution system would therefore improve the performance of our pipeline beyond the level of the PMB.

7 Conclusion

We have presented a pipeline for converting Universal Dependencies parses into semantic representations in the form of DRSs. Our rule-based system is intended to provide as linguistically broad a coverage as possible, producing semantic structures which faithfully capture the relations between discourse referents encoded in syntactic structure. Sometimes the UD parse itself is the cause of friction, and we have suggested some ways in which the UD framework might be improved so as to reduce the difficulty of semantic interpretation. Since our rule system produces underspecified DRSs, we also discussed one example post-processing step used to enhance and fully specify our representations. Comparison with the PMB shows that in terms of raw coverage we still have a way to go, but that our goal of capturing universal structural information is on the right track, insofar as our rules seem to generalise across the four languages represented in the PMB to similar extents. Since we now have a successfully implemented system and a working framework for evaluation, we have laid the groundwork for further progress to be made on a theoretical level with regard to improving and expanding the coverage of our ruleset.

Acknowledgements

This research was funded by the Norwegian Research Council, project 300495 *Universal Natural Language Understanding*.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Lasha Abzianidze, Rik van Noord, Hessel Haagsma, and Johan Bos. 2019. The first shared task on discourse representation structure parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*. Association for Computational Linguistics.

Ash Asudeh. 2022. Glue Semantics. *Annual Review of Linguistics*, 8:321–341.

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, Online. Association for Computational Linguistics.

Lucas Champollion. 2015. The interaction of compositional semantics and event semantics. *Linguistics and Philosophy*, 38(1):31–66.

Haskell B. Curry and Robert Feys. 1958. *Combinatory logic: volume I*. North Holland, Amsterdam.

Mary Dalrymple, John Lamping, and Vijay Saraswat. 1993. LFG semantics via constraints. In Steven Krauwer, Michael Moortgat, and Louis des Tombe, editors, *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics (EACL 1993)*, pages 97–105.

Donald Davidson. 1967. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*, pages 81–120. University of Pittsburgh Press, Pittsburgh, PA.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press, Cambridge, MA.

Jamie Y. Findlay and Dag T. T. Haug. 2021. How useful are enhanced Universal Dependencies for semantic interpretation? In *Proceedings of the Sixth International Conference on Dependency Linguistics (Depling, SyntaxFest 2021)*, pages 22–34, Sofia, Bulgaria. Association for Computational Linguistics.

Dan Garrette and Ewan Klein. 2009. An extensible toolkit for computational semantics. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 116–127, Tilburg, The Netherlands. Association for Computational Linguistics.

Mozhdeh Gheini, Xiang Ren, and Jonathan May. 2021. Cross-attention is all you need: Adapting pretrained Transformers for machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1765, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jean-Yves Girard. 1987. Linear logic. *Theoretical Computer Science*, 50(1):1–102.

Matthew Gotham and Dag T. T. Haug. 2018. Glue semantics for Universal Dependencies. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG'18 Conference*, pages 208–226. CSLI Publications, Stanford, CA.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2022. Survey of low-resource machine translation. *Computational Linguistics*, 48(3):673–732.

Dag T. T. Haug and Jamie Y. Findlay. 2023. Formal semantics for Dependency Grammar. In *Proceedings of the Seventh International Conference on Dependency Linguistics (Depling 2023)*. Association for Computational Linguistics.

William A. Howard. 1980. The formulae-as-types notion of construction. In *To H. B. Curry: essays on combinatory logic, lambda calculus, and formalism*, pages 479–490. Academic Press, London. Circulated in unpublished form from 1969.

Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kuebler. 2020. MonaLog: a lightweight system for natural language inference based on monotonicity. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 334–344, New York, New York. Association for Computational Linguistics.

Aikaterini-Lida Kalouli and Richard Crouch. 2018. GKR: the graphical knowledge representation for semantic parsing. In *Proceedings of the Workshop on Computational Semantics beyond Events and Roles*, pages 27–37, New Orleans, Louisiana. Association for Computational Linguistics.

Hans Kamp and Uwe Reyle. 1993. *From discourse to logic*. Kluwer, Dordrecht.

Hans Kamp, Josef van Genabith, and Uwe Reyle. 2011. Discourse Representation Theory. In Dov M. Gabbay and Franz Guenthner, editors, *Handbook of philosophical logic*, 2nd edition, volume 15, pages 125–394. Springer, Berlin.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of English verbs. *Language Resources and Evaluation*, 42(1):21–40.

Jiangming Liu, Shay B. Cohen, Mirella Lapata, and Johan Bos. 2021. Universal discourse representation structure parsing. *Computational Linguistics*, 47(2):445–476.

Moritz Messmer and Mark-Matthias Zymla. 2018. The Glue Semantics Workbench: a modular toolkit for exploring linear logic and Glue Semantics. In *Proceedings of the LFG'18 Conference*, pages 249–263, Stanford, CA. CSLI Publications.

Joakim Nivre. 2016. Universal Dependencies: A cross-linguistic perspective on grammar and lexicon. In *Proceedings of the Workshop on Grammar and Lexicon: interactions and interfaces (GramLex)*, pages 38–40, Osaka, Japan. The COLING 2016 Organizing Committee.

Joakim Nivre, Paola Marongiu, Filip Ginter, Jenna Kanerva, Simonetta Montemagni, Sebastian Schuster, and Maria Simi. 2018. Enhancing Universal Dependency treebanks: A case study. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 102–107, Brussels, Belgium. Association for Computational Linguistics.

Jenna Nyblom, Samuel Kohonen, Katri Haverinen, Tapio Salakoski, and Filip Ginter. 2013. Predicting conjunct propagation and other extended Stanford dependencies. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 252–261, Prague, Czech Republic. Charles University in Prague, Matfyzpress, Prague, Czech Republic.

Terence Parsons. 1990. *Events in the semantics of English: a study in subatomic semantics*. MIT Press, Cambridge, MA.

Agnieszka Patejuk and Adam Przepiórkowski. 2018. *From Lexical Functional Grammar to Enhanced Universal Dependencies: linguistically informed treebanks of Polish*. Institute of Computer Science Polish Academy of Sciences, Warsaw.

Adam Przepiórkowski and Agnieszka Patejuk. 2018. Arguments and adjuncts in Universal Dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3837–3852, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Adam Przepiórkowski and Agnieszka Patejuk. 2020. From Lexical Functional Grammar to enhanced Universal Dependencies: the UD-LFG treebank of Polish. *Language Resources and Evaluation*, 54(1):185–221.

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*.

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, Portorož, Slovenia. European Language Resources Association (ELRA).

Rik van Noord. 2022. Rikvn/drs_parsing: Scripts to evaluate scoped meaning representations. `https://github.com/RikVN/DRS_parsing`. Accessed: 2022-07-19.

Rik van Noord, Antonio Toral, and Johan Bos. 2020. Character-level representations improve DRS-based semantic parsing even in the age of BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4587–4603, Online. Association for Computational Linguistics.

Noortje J. Venhuizen, Johan Bos, and Harm Brouwer. 2013. Parsimonious semantic representations with projection pointers. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 252–263, Potsdam, Germany. Association for Computational Linguistics.

Noortje J. Venhuizen, Johan Bos, Petra Hendriks, and Harm Brouwer. 2018. Discourse semantics with information structure. *Journal of Semantics*, 35(1):127–169.

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

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.