Transparent Semantic Parsing with Universal Dependencies using Graph Transformations

Wessel Poelman, Rik van Noord, Johan Bos
Center for Language and Cognition
University of Groningen
The Netherlands
contact@wesselpoelman.nl
{r.i.k.van.noord,johan.bos}@rug.nl

Abstract

Even though many recent semantic parsers are based on deep learning methods, we should not forget that rule-based alternatives might offer advantages over neural approaches with respect to transparency, portability, and explainability. Taking advantage of existing off-the-shelf Universal Dependency parsers, we present a method that maps a syntactic dependency tree to a formal meaning representation based on Discourse Representation Theory. Rather than using lambda calculus to manage variable bindings, our approach is novel in that it consists of using a series of graph transformations. The resulting UD semantic parser shows good performance for English, German, Italian and Dutch, with F-scores over 75%, outperforming a neural semantic parser for the lower-resourced languages. Unlike neural semantic parsers, our UD semantic parser does not hallucinate output, is relatively easy to port to other languages, and is completely transparent.

1 Introduction

Semantic parsing is the task of mapping natural language sentences to a formal meaning representation such as Abstract Meaning Representations (Banarescu et al., 2013) or Discourse Representation Structures (Bos et al., 2017). The current trend in this area is strongly geared towards using methods based on deep learning. The best performing parsers use pre-trained language models (van Noord et al., 2020; Zhou et al., 2021; Bevilacqua et al., 2021; Bai et al., 2022). But a good performance is perhaps not the only thing that matters. A drawback of neural semantic parsers is that their output lacks explainability: why are the meaning representations composed in the way they are? Moreover, they require vast amounts of training data, and are usually specific for a particular language. In addition, their performance usually decreases for longer input sentences.

In other words, it may look like we have made a lot of progress, but viewed from a different perspective, we might actually have made a step back. This is especially so with regards to transparency and interpretability of semantic parsers. In this paper we describe a semantic parsing system for Discourse Representation Structures — the formal meaning representations proposed by Discourse Representation Theory (Kamp and Reyle, 1993; Abzianidze et al., 2017) — that is based on Universal Dependencies (UD, de Marneffe et al., 2021). The first advantage of the UD framework is that it has been developed for numerous languages (using a cross-linguistically consistent annotation scheme) and that several high-performing parsers have been developed for UD. This will make it easier, as we will show, to develop semantic parsers for languages other than English, in our case German, Italian and Dutch. The second advantage of using UD as input describing the syntactic structure of the sentence, is that it provides us with explainable support of the output of the meaning representation, based on the derivation provided by the UD parser.

The innovative contribution of the system, *UD-boxer*, that we describe here is in the way the meaning representations are computed. Even though the original Discourse Representation Structure (DRS) is formally an ordered pair of a set of discourse referents and a set of conditions, we recast the DRS as a directed acyclic graph. Through graph transformation rules our system changes the input UD syntactic representation step-by-step into a fully fledged formal meaning representation.

2 Related Work

Our aims are similar to those of Reddy et al. (2016) and Reddy et al. (2017), who map UD to logical forms in three steps: (1) enriching the UD tree with missing syntactic information and long-distance dependencies, turning it into a graph, (2) binarization of the dependency graph, (3) substituting the

Proceedings of the 29th International Conference on Computational Linguistics, pages 4186–4192
October 12–17, 2022.

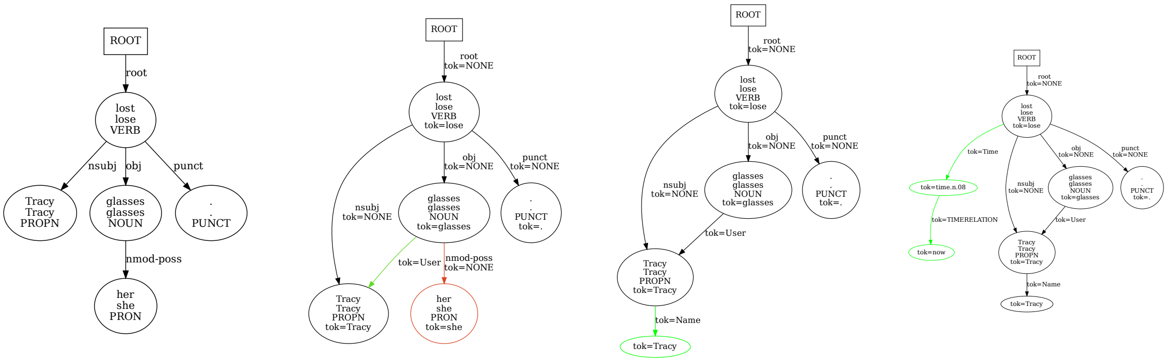

Figure 1: Graph transformations for *Tracy lost her glasses*, from left to right: initial UD graph, connecting the User role, expanding the proper name, and adding and connecting tense information. The *token* attribute refers to the semantic label or concept for a given node or edge.

words and using typed lambda expressions that encode the lexical semantics and dependency labels for λ-expressions that either copy, invert or merge lambda-expression to compose predicate-argument structures, and (4) applying β-conversion to get a reduced, normalized logical form.

The main difference of our work with that of Reddy et al. (2017) is that we do not require complicated operations involving logical variables. By making the target meaning representations free of variables in the form of a graph, the mapping from UD to meaning representation is solely based on a sequence of graph transformations (Zhizhkun, 2006). This allows us to apply our method on languages other than English.

Similar in spirit to our work, but different in execution, is recent work by Shen and Evang (2022), who present a DRS parser that is competitive in accuracy with recent sequence-to-sequence models and at the same time *compositional*. This latter property makes their system transparent and more explainable. Shen and Evang (2022) recast DRS parsing as a sequence labeling task, and achieve a good performance with F-scores of 84.4 for English, 78.3 for German, 80.4 for Italian, and 72.1 for Dutch on PMB 3.0.0 data (and therefore not directly comparable with our results, working with a more recent version of the PMB).

3 Method

3.1 Overall Idea

Our semantic parsing method is based on the insight provided by Bos (2021) that Discourse Representation Structures can be represented as relatively simple directed acyclic graphs without resorting to variables. In a Discourse Representation Graph (DRG) the nodes denote entities (represented by a WordNet synset) and constants (names, numbers, dates, etc.), the edges denote thematic roles, comparison operators, and discourse relations. Our semantic mapping comprises a series of transformations from UD to DRG, exemplified by Figure 1 and Figure 2. Our full system is publicly available: https://github.com/WPoelman/ud-boxer

We use semantically annotated data from the Parallel Meaning Bank (Abzianidze et al., 2017). The PMB provides a large set of English, German, Italian and Dutch sentences paired with Discourse Representation Structures (DRSs). Since release 4.0.0[1] the PMB also provides DRSs in a variable-free variant, using relative indices instead of variables, following the procedure outlined in Bos (2021). This notation corresponds directly to a DRG. There is a straightforward mapping from DRG to DRS of formulas of first-order logic (see Figure 2). We use this format in developing our system. The PMB has gold, silver (partially annotated) and bronze (no manual annotations) standard data available. We only use gold for UD-boxer, while for Neural Boxer, a strong neural parser based on van Noord et al. (2020), we use all available data for German, Italian and Dutch. However, for training Neural Boxer on English data we only use the gold and silver data (van Noord et al., 2018). The data splits are shown in Table 1. Note that English also has an extra evaluation set of 830 instances. This was the hidden test set in the shared task of Abzianidze et al. (2019) and now serves as an extra test set.

[1] The data of the Parallel Meanng Bank is available here: https://pmb.let.rug.nl/data.php.

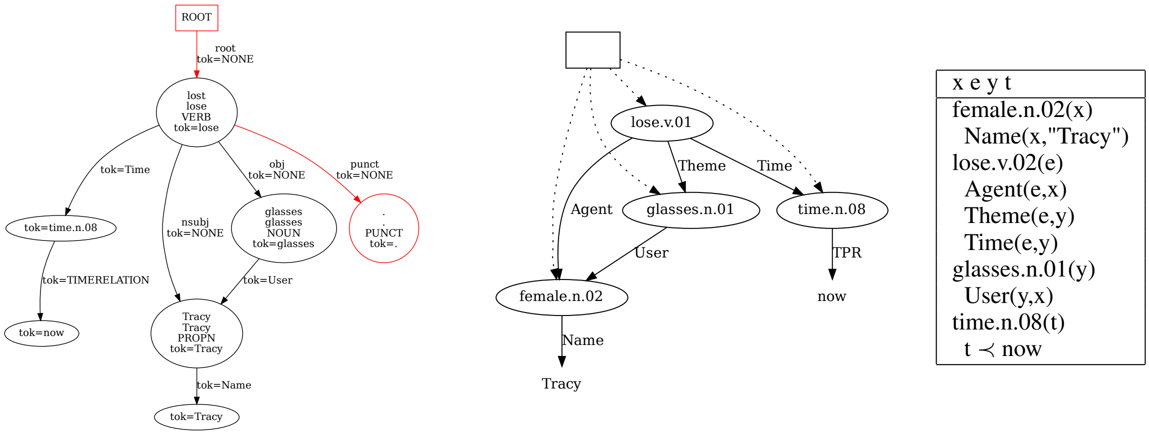

Figure 2: Removing redundant nodes followed by substitution semantic symbols for syntactic relations, and corresponding DRS in box format for *Tracy lost her glasses*.

	Gold			Silver	Bronze
	Train	Dev	Test	Train	Train
English	7,668	1,169	1,048	127,303	151,493
German	1,738	559	547	6,355	156,286
Italian	685	540	461	4,088	100,963
Dutch	539	491	437	1,440	28,265

Table 1: Number of documents for the four languages for PMB release 4.0.0.

3.2 System Overview

The overall system expects a sentence as input and consists of three main steps: (1) creating a UD parse; (2) applying the graph transformation rules; and (3) substituting syntactic labels for semantic entities. The final output is a DRG and can be exported to various formats.

The first step is implemented by using existing off-the-shelf UD parsers. This is a modular part of the system — any UD parser can be plugged in. In the context of this paper we used two state-of-the-art UD parsers: Stanza (Qi et al., 2020) and Trankit (Nguyen et al., 2021), both of which go head to head in their performance for English and also achieve good results for the other languages of our study.

The second step, the graph transformation, is carried out by using GREW, a graph rewriting framework specifically designed for linguistic graphs and trees (Bonfante et al., 2018; Guillaume, 2021). The focus of this step is to apply *structural* changes to the graph: adding, removing or combining nodes and edges. Some node and edge label substitution might be carried out already during graph transformation, but most of that is left to the final step. Figure 3 shows a language-neutral transformation rule that connects a thematic role to an entity, whereas Figure 4 is an example of a language-specific rule.

```
rule connect_user {
    pattern {
        USER [upos=PROPN|NOUN];
        * -[1=nsubj]-> USER;
        REL: TARGET -[1=nmod, 2=poss]->
              INDICATOR;
    }
    without {
        TARGET -[token=User]-> USER;
    }
    commands {
        add_edge TARGET -[token=User]->
               USER;
        del_edge REL;
        del_node INDICATOR;
    }
}
```

Figure 3: Example of language-neutral rule (in GREW syntax) to connect the User role to an entity.

The final step, substitution, involves labeling nodes and edges that do not have valid DRS labels yet, as well as connecting box nodes (the only structural part of this step). Currently, this step is comprised of applying simple mappings extracted from the training data and leveraging syntactic and morphological information from the UD parse. This is also a modular component and additional approaches can be added. Existing systems that go beyond syntax are a good candidate to be added here, e.g., named entity recognition systems.

```
rule box_negation_det {
    pattern {
        N [lemma=no|not|never];
        * -[1=advmod|det]-> N;
    }
    without {
        P [token=NEGATION];
    }
    commands {
        del_node N;

        add_node NEGATION_BOX;
        NEGATION_BOX.token = NEGATION;
    }
}
```

Figure 4: Example of English specific rule to introduce a negation box. The box gets connected in the substitution and labeling step.

The transformation rules as well as the mappings for the substitution are developed using the training data and tested on the development set. Currently there are 19 language independent rules and four specific rules per language. These specific rules deal with either negation or quantifiers (e.g. *all*, *every*, *none*). Rules were developed by analyzing the UD graph and gold SBN graph side-by-side for a given example sentence. We then aimed to create the most general and simple rule(s) that (structurally) transformed the UD graph into the SBN graph.

The node and edge mappings are extracted when the graph transformations are applied and result in a graph isomorphic to the gold-standard graph. The UD information is then extracted and stored per triple (from node, edge, to node) if the mapping was correct. This creates a positive feedback loop, as the rules improve, the labeling improves as well. This process was bootstrapped by creating a tiny set of initial mappings from dependency relations to DRS roles and operators. Our approach here serves as a baseline of sorts, since word sense disambiguation and edge labeling are only done with the most frequent occurrences in the training data.

3.3 Evaluating DR Graphs

Counter is the standard evaluation tool for DRSs (van Noord et al., 2018). However, it is specifically designed for the *clausal* notation of DRS. This notation does not use a graph-like structure directly, but works with clauses that can have three or four components. It is therefore not suitable for the (simpler) DRGs that our system produces.

SMATCH (Cai and Knight, 2013) was created for evaluating Abstract Meaning Representations, which are directed acyclic graphs, like DRGs. SMATCH supports the Penman notation (Kasper, 1989), converts a graph into a set of triples, while automatically performing role inversion when needed. By converting a DRG into Penman format, we can simply use SMATCH to compare system output with the gold standard, for which SMATCH computes an F-score based on matching triples. A DRG in Penman format is shown in Figure 5.

```
(b0 / box
    :member (e1 / entity
        :lemma female :pos n :sense 02
        :Name "Tracy")
    :member (e2 / entity
        :lemma lose :pos v :sense 02
        :Agent e1 :Theme e3 :Time e4)
    :member (e3 / entity
        :lemma glasses :pos n :sense 01
        :User e1)
    :member (e4 / entity
        :lemma time :pos n :sense 08
        :TPR now))
```

Figure 5: Penman format for a Discourse Representation Graph for *Tracy lost her glasses.*

As Figure 5 shows, we split up WordNet synsets components to support a more fine-grained evaluation. This also gives us flexibility in the evaluation process, where we can toggle between evaluating word sense disambiguation (strict) or not (lenient). In this paper, we use only strict evaluation.

3.4 Comparison System

For comparison with our system we train a neural DRG parser based on BERT (Devlin et al., 2019), following van Noord et al. (2020).[2] This is a bi-LSTM sequence-to-sequence model, which uses (frozen) BERT embeddings to initialize the encoder. Specifically, we use `bert-base-cased` for English and `bert-base-multilingual-cased` for the other languages. The word-level decoder is trained from scratch. We do not apply any preprocessing nor postprocessing, simply taking the input sentence and output DRS in sequential box notation as is. We follow the procedure described in van Noord et al. (2020) by first pretraining on gold + non-gold data, after which we fine-tune on just the gold data.

[2] Detailed instructions can be found here: https://github.com/RikVN/Neural_DRS/blob/master/AllenNLP.md#sbn-experiments.

	English		German		Italian		Dutch	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
UD-Boxer (Stanza)	82.1 (0.3)	82.0 (0.0)	**78.4 (0.0)**	77.3 (0.0)	76.2 (1.9)	78.4 (0.9)	75.5 (0.0)	**75.8 (0.0)**
UD-Boxer (Trankit)	81.9 (0.3)	81.8 (0.0)	**78.4 (0.0)**	**77.5 (0.0)**	**77.8 (0.0)**	**79.1 (0.0)**	**75.8 (0.0)**	**75.8 (0.0)**
Neural Boxer (gold)	82.8 (4.6)	84.0 (3.7)	64.2 (0.4)	63.8 (0.2)	55.5 (1.5)	55.7 (1.5)	51.2 (0.2)	51.1 (0.4)
Neural Boxer (best)	**92.5 (2.0)**	**92.5 (2.3)**	74.6 (0.4)	74.7 (0.5)	75.6 (0.0)	75.4 (0.0)	71.9 (0.9)	71.6 (1.0)

Table 2: Average macro F1-scores on the dev and test set of the four languages in PMB 4.0.0. The number in parentheses indicates the percentage of ill-formed DRSs in the output. For Neural Boxer, *best* indicates that it was trained on gold, silver and bronze data (German, Italian, and Dutch) or only on gold and silver data (English). UD-Boxer (Stanza) and UD-Boxer (Trankit) obtain an F-score of 81.3 and 81.5 on the English *evaluation* set.

4 Results

Table 2 shows the main results for UD-boxer. We show the results of using two syntactic parsers (Stanza and Trankit) and compare to the performance of the neural system, trained on just the gold PMB data and on gold and non-gold data. Our system is not competitive with the best Neural Boxer for English, but clearly outperforms this model for German, Italian and Dutch. However, when only using gold data, the performance of our model is quite close to Neural Boxer for English, while producing considerably fewer ill-formed DRSs.

In a manual analysis, we found that the few errors made by UD-Boxer were all caused by unexpected sentence roots in the UD output of the Stanza parser. A case in point is the input *All of my friends like computer games* where Stanza decides that it is a noun phrase with *All* as the root, whereas Trankit assigns *like* as the root. Currently no transformation rules deal with such cases. The graph gets malformed because the root is cut away at some point since it is a determiner and those can generally be left out of DRSs.

But the majority of ill-formed output is produced by the neural parsers. An example, which also exhibits hallucination of semantic information, is the output DRG for *She died of tuberculosis* (Figure 6), where the disease changed into a heart attack and is recognized as an anonymous geological formation (there is a role Name without a constant). This strange phenomenon occurs often in output from both neural parsers. The problem with detecting these anomalies is that often these graphs score well, even though they are semantically ill-formed.

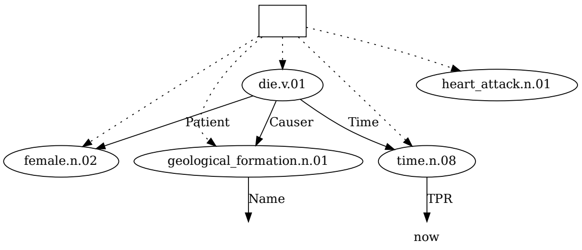

Figure 6: Erroneously (wrong type of disease, incorrect named entity) and ill-formed output (node missing) of Neural Boxer for *She died of tuberculosis.*

5 Conclusion

Our results show competitive performance between UD-Boxer and Neural Boxer for English, when a limited amount of training data is available for the neural approach. In addition, it shows strong cross-lingual performance using a few simple language-specific rules and only gold training data to extract the mappings. Adding more transformation rules and creating a label substitution component based on machine learning will likely push the performance of UD-Boxer even higher. The phenomena we have in mind are named entities, numeral expressions, time and date expressions, and discourse relations, for which only simple rules have been defined so far. Since UD-Boxer is a modular system, it is rather straightforward to add such rules.

An important difference between the two semantic parsers is that UD-Boxer is guaranteed to output a well-formed meaning representation, provided that the input UD is accurate. Because Neural Boxer is following a seq2seq transformer approach, the output does not always correspond to a graph and therefore requires ad-hoc postprocessing rules to make such output interpretable. Another serious deficiency of neural parsers is that it sometimes hallucinates semantic material without warning. A transparent semantic parser, even with a slightly lower performance in some cases, might be a good alternative for certain applications, in particular when lower-resourced languages are involved.

Acknowledgements

The Tesla K40 GPU used in this work was kindly donated to us by the NVIDIA Corporation. We thank the Center for Information Technology of the University of Groningen for providing access to the Peregrine high performance computing cluster. We also thank the three anonymous reviewers for their valuable feedback.

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*, Gothenburg, Sweden. Association for Computational Linguistics.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

G. Bonfante, B. Guillaume, and G. Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*. Wiley.

Johan Bos. 2021. Variable-free discourse representation structures. *Semantics Archive*.

Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. 2017. The Groningen Meaning Bank. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, volume 2, pages 463–496. Springer.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bruno Guillaume. 2021. Graph Matching and Graph Rewriting: GREW tools for corpus exploration, maintenance and conversion. In *EACL 2021 - 16th conference of the European Chapter of the Association for Computational Linguistics*, Kiev/Online, Ukraine.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.

Robert T. Kasper. 1989. A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics.

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, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

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.

Minxing Shen and Kilian Evang. 2022. DRS parsing as sequence labeling. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 213–225, Seattle, Washington. Association for Computational Linguistics.

Rik van Noord, Lasha Abzianidze, Hessel Haagsma, and Johan Bos. 2018. Evaluating scoped meaning representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 1685–1693. European Language Resources Association (ELRA).

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018. Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

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. Association for Computational Linguistics.

Valentin Zhizhkun. 2006. *Graph Transformations for Natural Language Processing*. Ph.D. thesis, University of Amsterdam.

Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, Young-Suk Lee, Radu Florian, and Salim Roukos. 2021. Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based AMR parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6290, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.