

Exploring Neural Methods for Parsing Discourse Representation Structures

Rik van Noord Lasha Abzianidze Antonio Toral Johan Bos

Center for Language and Cognition, University of Groningen
{r.i.k.van.noord, l.abzianidze, a.toral.ruiz, johan.bos}@rug.nl

Abstract

Neural methods have had several recent successes in semantic parsing, though they have yet to face the challenge of producing meaning representations based on formal semantics. We present a sequence-to-sequence neural semantic parser that is able to produce Discourse Representation Structures (DRSs) for English sentences with high accuracy, outperforming traditional DRS parsers. To facilitate the learning of the output, we represent DRSs as a sequence of flat clauses and introduce a method to verify that produced DRSs are well-formed and interpretable. We compare models using characters and words as input and see (somewhat surprisingly) that the former performs better than the latter. We show that eliminating variable names from the output using De Bruijn indices increases parser performance. Adding silver training data boosts performance even further.

1 Introduction

Semantic parsing is the task of mapping a natural language expression to an interpretable meaning representation. Semantic parsing used to be the domain of symbolic and statistical approaches (Pereira and Shieber, 1987; Zelle and Mooney, 1996; Blackburn and Bos, 2005). Recently, however, neural methods, and in particular sequence-to-sequence models, have been successfully applied to a wide range of semantic parsing tasks. These include code generation (Ling et al., 2016), question answering (Dong and Lapata, 2016; He and Golub, 2016) and Abstract Meaning Representation parsing (Konstas et al., 2017). Because these models have no intrinsic knowledge of the

structure (tree, graph, set) they have to produce, recent work also focused on structured decoding methods, creating neural architectures that always output a graph or a tree (Alvarez-Melis and Jaakkola, 2017; Buys and Blunsom, 2017). These methods often outperform the more general sequence-to-sequence models but are tailored to specific meaning representations.

This paper will focus on parsing Discourse Representation Structures (DRSs) proposed in Discourse Representation Theory (DRT), a well-studied formalism developed in formal semantics (Kamp, 1984; Van der Sandt, 1992; Asher, 1993; Kamp and Reyle, 1993; Muskens, 1996; Van Eijck and Kamp 1997; Kadmon, 2001; Asher and Las-carides, 2003), dealing with many semantic phenomena: quantifiers, negation, scope ambiguities, pronouns, presuppositions, and discourse structure (see Figure 1). DRSs are recursive structures and thus form a challenge for sequence-to-sequence models because they need to generate a well-formed structure and not something that looks like one but is not interpretable.

The problem that we try to tackle bears similarities to the recently introduced task of mapping sentences to an Abstract Meaning Representation (AMR; Banarescu et al. 2013). But there are notable differences between DRS and AMR. Firstly, DRSs contain scope, which results in a more linguistically motivated treatment of modals, quantification, and negation. Secondly, DRSs contain a substantially higher number of variable bindings (**reentrant nodes** in AMR terminology), which are challenging for learning (Damonte et al., 2017).

DRS parsing was attempted in the 1980s for small fragments of English (Johnson and Klein, 1986; Wada and Asher, 1986). Wide-coverage

Raw input:

Tom isn't afraid of anything.

System output of a DRS in a clausal form:

```

b1 REF x1           b3 REF s1
b1 male "n.02" x1   b3 Time s1 t1
b1 Name x1 "tom"    b3 Experiencer s1 x1
b2 REF t1           b3 afraid "a.01" s1
b2 EQU t1 "now"     b3 Stimulus s1 x2
b2 time "n.08" t1   b3 REF x2
b0 NOT b3           b3 entity "n.01" x2

```

The same DRS in a box format:

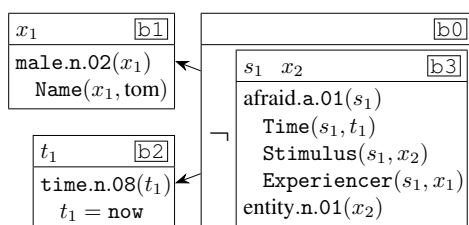


Figure 1: DRS parsing in a nutshell. Given a raw text, a system has to generate a DRS in the clause format, a flat version of the standard box notation. The semantic representation formats are made more readable by using various letters for variables: the letters x , e , s , and t are used for discourse referents denoting individuals, events, states, and time, respectively, and b is used for variables denoting DRS boxes.

DRS parsers based on supervised machine learning emerged later (Bos, 2008b; Le and Zuidema, 2012; Bos, 2015; Liu et al., 2018). The objectives of this paper are to apply neural methods to DRS parsing. In particular, we are interested in answers to the following research questions (RQs):

1. Are sequence-to-sequence models able to produce formal meaning representations (DRSs)?
2. What is better for input: sequences of characters or sequences of words; does tokenization help; and what kind of casing is best used?
3. What is the best way of dealing with variables that occur in DRSs?
4. Does adding silver data increase the performance of the neural parser?
5. What parts of semantics are learned and what parts of semantics are still challenging?

We make the following contributions to semantic parsing:¹ (a) The output of our parser consists of interpretable scoped meaning representations,

¹The code is available here: https://github.com/RikVN/Neural_DRS.

guaranteed by a specially designed checking tool (§3). (b) We compare different methods of representing input and output in §4. (c) We show in §5 that using additional, non-gold standard data can improve performance. (d) We perform a thorough analysis of the produced output and compare our methods with symbolic/statistical approaches (§6).

2 Discourse Representation Structures

2.1 The Structure of DRS

DRSs are meaning representations introduced by DRT (Kamp and Reyle, 1993). In general, a DRS can be seen as an ordered pair $\langle \mathcal{A}, l : B \rangle$, where \mathcal{A} is a set of presuppositional DRSs, and B a DRS with a label l . The presuppositional DRSs \mathcal{A} can be viewed as propositions that need to be anchored in the context in order to make the main DRS B true, where presuppositions comprise anaphoric phenomena, too (Van der Sandt, 1992; Geurts, 1999; Beaver, 2002).

DRSs are either elementary DRSs or segmented DRSs. An elementary DRS is an ordered pair of a set of discourse referents and a set of conditions. There are basic conditions and complex conditions. A basic condition is a predicate applied to constants or discourse referents, whereas a complex condition can introduce Boolean operators ranging over DRSs (negation, conditionals, disjunction). Segmented DRSs capture discourse structure by connecting two units of discourse by a discourse relation (Asher and Lascarides, 2003).

2.2 Annotated Corpora

Despite a long tradition of formal interest in DRT, it is only recently that textual corpora annotated with DRSs have been made available. The Groningen Meaning Bank (GMB) is a large corpus with DRS annotation for mostly short English newspaper texts (Basile et al., 2012; Bos et al., 2017). The DRSs in this corpus are produced by an existing semantic parser and then partially corrected. The DRSs in the GMB are therefore not gold standard.

A similar corpus is the Parallel Meaning Bank (PMB), which provides DRSs for English, German, Dutch, and Italian sentences based on a parallel corpus (Abzianidze et al., 2017). The PMB, too, is constructed using an existing semantic parser, but a part of it is completely manually checked and corrected (i.e., gold standard). In contrast to the GMB, the PMB involves two major

additions: (a) its semantics are refined by modeling tense and using semantic tagging (Bjerva et al., 2016; Abzianidze and Bos, 2017), and (b) the non-logical symbols of the DRSs corresponding to concepts and semantic roles are grounded in WordNet (Fellbaum, 1998) and VerbNet (Bonial et al., 2011), respectively.

These additions make the DRSs of the PMB more fine-grained meaning representations. For this reason we choose the PMB (over the GMB) as our corpus for evaluating our semantic parser. Even though the sentences in the current release of the PMB are relatively short, they contain many difficult semantic phenomena that a semantic parser has to deal with: pronoun resolution, quantifiers, scope of modals and negation, multi-word expressions, word senses, semantic roles, presupposition, tense, and discourse relations. As far as we know, we are the first group to use the PMB corpus for semantic parsing.

2.3 Formatting DRSs with Boxes and Clauses

The usual way to represent DRSs is the well-known box format. To facilitate reading a DRS with unresolved presuppositions, it can be depicted as a network of boxes, where a non-presuppositional (i.e., main) DRS $l : B$ is connected to the presuppositional DRSs \mathcal{A} with arrows. Each box comes with a unique label and has two rows. In the case of elementary DRSs, these rows contain discourse referents in the top row and conditions in the bottom row (Figure 1). A segmented DRS has a row with labeled DRSs and a row with discourse relations (Figure 2).

The DRS in Figure 1 consists of a main box b_0 and two presuppositional boxes, b_1 and b_2 . Note that b_0 has no discourse referents but introduces negation via a single condition $\neg b_3$ with a nested box b_3 . The conditions of b_3 represent unary and binary relations over discourse referents that are introduced either by b_3 or the presuppositional DRSs.

A clausal form is another way of formatting DRSs. It represents a DRS as a set of clauses (see Figures 1 and 2). This format is better suited for machine learning than the box format, as it has a simple, flat structure and facilitates partial matching of DRSs, which is useful for evaluation (van Noord et al., 2018). Conversion from the box notation to the clausal form and vice versa is transparent: Discourse referents, conditions, and dis-

00/3008: He played the piano and she sang.

b0 DRS b1	b0 DRS b5
b2 REF x1	b6 REF x3
b2 male "n.02" x1	b6 female "n.02" x3
b1 REF e1	b5 REF e2
b1 play "v.03" e1	b5 sing "v.01" e2
b1 Agent e1 x1	b5 Agent e2 x3
b1 Theme e1 x2	b5 Time e2 t2
b3 REF x2	b7 REF t2
b3 piano "n.01" x2	b7 TPR t2 "now"
b4 REF t1	b7 time "n.08" t2
b4 time "n.08" t1	b0 CONTINUATION b1 b5
b4 TPR t1 "now"	b1 Time e1 t1

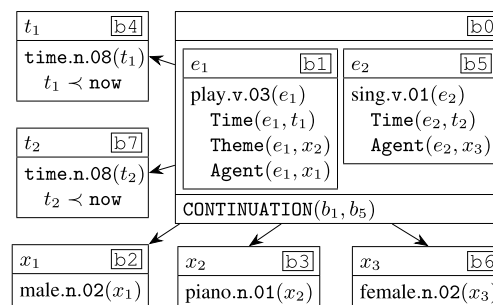


Figure 2: A segmented DRS. Discourse relations are formatted with uppercase characters.

course relations in the clausal form are preceded by the label of the box in which they occur. Notice that the variable letters in the semantic representations are automatically set and they simply serve for readability purposes. Throughout the experiments described in this paper, we utilize clausal form DRSs.

3 Method

3.1 Annotated Data

We use the English DRSs from release 2.1.0 of the PMB (Abzianidze et al., 2017).² The release suggests using the parts 00, 10, 20, and 30 as the development set, resulting in 3,998 training and 557 development instances. Basic statistics are shown in Table 1, and the number of occurrences of some of the semantic phenomena mentioned in §2.2 are given in Table 2.

Because this is a rather small training set, we tune our model using 10-fold cross-validation (CV) on the training set, rather than tuning on a separate development set. This means that we will use the suggested development set as a test set (and refer to it as such). When testing on this set, we train a model on all available training data. The utilized PMB release also comes with “silver” data—namely, 71,308 DRSs that are only partially

²<http://pmb.let.rug.nl/data.php>.

	Sentences	Tokens	Avg tok/sent
Gold train	3,998	24,917	6.2
Gold test	557	3,180	5.7
Silver	73,778	638,610	8.7

Table 1: Number of documents, sentences, and tokens for the English part of PMB release 2.1.0. Note that the number of tokens is based on the PMB tokenization, treating multiword expressions as a single token.

Phenomenon	Train	Test	Silver
Negation & modals	442	73	17,527
Scope ambiguity	≈67	15	≈3,108
Pronoun resolution	≈291	31	≈3,893
Discourse rel. & imp.	254	33	16,654
Embedded clauses	≈160	30	≈46,458

Table 2: Counts of relevant semantic phenomena for PMB release 2.1.0.³ These phenomena are described and further discussed in §6.3.

manually corrected. In addition, we use the DRSs from the silver data but without the manual corrections, which makes them “bronze” DRSs (following PMB terminology). Our experiments will initially use only the gold standard data, after which we will use the silver or bronze data to further push the score of our best systems.

3.2 Clausal Form Checker

The clausal form of a DRS needs to satisfy a set of constraints in order to correspond to a semantically interpretable DRS, that is, translatable into a first-order logic formula without free occurrences of a variable (Kamp and Reyle, 1993). For example, all discourse referents need to be explicitly introduced with a REF clause to avoid free occurrences of variables.

We implemented a clausal form checker that validates the clausal form if and only if it represents a semantically interpretable DRS. Distinguishing box variables from entity variables is crucial for the validity checking, but automatically learned clausal forms are not expected to differentiate variable types. First, the checker separately

³The phenomena are automatically counted based on clausal forms. The counting algorithm does not guarantee the exact number for certain phenomena, though it returned the exact counts of all the phenomena on the test data except the pronoun resolution (30).

parses each clause in the form to induce variable types based on the fixed set of comparison and DRS operators. After typing all the variables, the checker verifies whether the clauses collectively correspond to a DRS with well-formed semantics. For each box variable in a discourse relation, existence of the corresponding box inside the same segmented DRS is checked. For each entity variable in a condition, an introduction of the binder (i.e., accessible) discourse variable is found. The goal of these two steps is to prevent free occurrences of variables in DRSs. While binding the entity variables, necessary accessibility relations between the boxes are induced. In the end, the checker verifies the transitive closure of the induced accessibility relation on loops and checks existence of a unique main box of the DRS.

The checker is applied to every automatically obtained clausal form. If a clausal form fails the test, it is considered as ill-formed and will not have a single clause matched with the gold standard when calculating the F-score.

3.3 Evaluation

A DRS parser is evaluated by comparing its output DRS to a gold standard DRS using the Counter tool (van Noord et al., 2018). Counter calculates an F-score over matching clauses. Because variable names are meaningless, obtaining the matching clauses essentially is a search for the best variable mapping between two DRSs. Counter tries to find this mapping by performing a hill-climbing search with a predefined number of restarts to avoid getting stuck in a local optimum, which is similar to the evaluation system SMATCH (Cai and Knight, 2013) for AMR parsing.⁴ Counter generalizes over WordNet synsets (i.e., a system is not penalized for predicting a word sense that is in the same synset as the gold standard word sense).

To calculate whether there is a significant difference between two systems, we perform approximate randomization (Noreen, 1989) with $\alpha = 0.05$, $R = 1,000$, and $F(model_1) > F(model_2)$ as test statistics for each individual DRS pair.

3.4 Neural Architecture

We use a recurrent sequence-to-sequence neural network (henceforth seq2seq) with two

⁴Counter ignores REF clauses in the calculation of the F-score because they are usually redundant and therefore inflate the final score (van Noord et al., 2018).

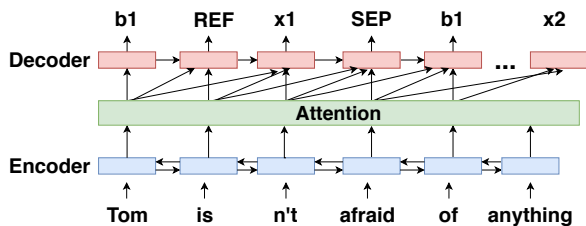


Figure 3: The sequence-to-sequence model with word-representation input. SEP is used as a special character to separate clauses in the output.

bidirectional long short-term memory (LSTM) layers and 300 nodes, implemented in OpenNMT (Klein et al., 2017). The network encodes a sequence representation of the natural language utterance, while the decoder produces the sequences of the meaning representation. We apply dropout (Srivastava et al., 2014) between both the recurrent encoding and decoding layers to prevent overfitting, and use general attention (Luong et al., 2015) to selectively give more weight to certain parts of the input sentence. An overview of the general framework of the seq2seq model is shown in Figure 3.

During decoding we perform beam search with length normalization, which in neural machine translation (NMT) is crucial to obtaining good results (Britz et al., 2017). We experimented with a wide range of parameter settings, of which the final settings can be found in Table 3.

We opted against trying to find the best parameter settings for each individual experiment (next to impossible in terms of computing time necessary, as a single 10-fold CV experiment takes 12 hours on GPU), but selected parameter settings that showed good performance for both the initial character and word-level representations (see §4 for details). The parameter search was performed using 10-fold CV on the training set. Training is stopped when there is no more improvement in perplexity on the validation set, which in our case occurred after 13–15 epochs.

A powerful, well-known technique in the field of NMT is to use an ensemble of models during decoding (Sutskever et al., 2014; Sennrich et al., 2016a). The resulting model averages over the predictions of the individual models, which can balance out some of the errors. In our experiments, we apply this method when decoding on the test set, but not for our experiments of 10-fold CV (this would take too much computation time).

Parameter	Value	Parameter	Value
RNN-type	LSTM	dropout	0.2
encoder-type	brnn	dropout type	naive
optimizer	sgd	bridge	copy
layers	2	learning rate	0.7
nodes	300	learning rate decay	0.7
min freq source	3	max grad norm	5
min freq target	3	beam size	10
vector size	300	length normalization	0.9

Table 3: Parameters explored during training and testing with their final values. All other parameters have default values.

4 Experiments with Data Representations

This section describes the experiments we conduct regarding the data representations of the input (English sentences) and output (a DRS) during training.

4.1 Between Characters and Words

We first try two (default) representations: character-level and word-level. Most semantic parsers use word-level representations for the input, but as a result are often dependent on pre-trained word embeddings or anonymization of the input⁵ to obtain good results. Character-level models avoid this issue but might be at a higher risk of producing ill-formed output.

Character-based model In the character-level model, the input (an English sentence) is represented as a sequence of individual characters. The output (a DRS in clause format) is linearized, with special characters indicating spaces and clause separators. The semantic roles (e.g., Agent, Theme), DRS operators (e.g., REF, NOT, POS), and deictic constants (e.g., "now", "speaker", "hearer") are not represented as character sequences, but treated as compound characters, meaning that REF is not treated as a sequence of R, E and F, but directly as REF. All proper names, WordNet senses, time/date expressions, and numerals are represented as character sequences.

⁵This is done to keep the vocabulary small. An example is to change all proper names to NAME in both the sentence and meaning representation during training. When producing output, the original names are restored by switching NAME with a proper name found in the input sentence (Konstas et al., 2017).

Word-based model In the word-level model, the input is represented as a sequence of words, using spaces as a separator (i.e., the original words are kept). The output is the same as for the character-based model, except that the character sequences are represented as words. We use pre-trained GloVe embeddings (Pennington et al., 2014)⁶ to initialize the encoder and decoder representations. In the DRS representation, there are semantic roles and DRS operators that might look like English words, but should not be interpreted as such (e.g. Agent, NOT). These entities are removed from the set of pre-trained embeddings, so that the model will learn them from scratch (starting from a random initialization).

Hybrid representations: BPE We do not necessarily have to restrict ourselves to using only characters or words as input representation. In NMT, byte-pair encoding (BPE; Sennrich et al. 2016b) is currently the de facto standard (Bojar et al., 2017). This is a frequency-based method that automatically finds a representation that is between character-level and word-level. It starts out with the character-level format and then does a predefined number of merges of frequently co-occurring characters. Tuning this number of merges determines whether the resulting representation is closer to character-level or word-level. We explore a large range of merges (1k–100k), while applying a corresponding set of pre-trained BPE embeddings (Heinzerling and Strube, 2018). However, none of the BPE experiments improved on the character-level or word-level score (F-scores between 57 and 68), only coming close when using a small number of merges (which is very close to character-level anyway). Therefore this technique was disregarded for further experiments.

Combined char and word There is also a fourth possible representation of the input: concatenating the character-level and word-level representations. This is uncommon in NMT because of the large size of the embedding space (hence their preference for BPE), but possible here since the PMB data contain relatively short sentences. We simply add the word embedding vector after the sequence of character-embeddings for each

Model	Prec	Rec	F-score	% ill
Char	78.1	69.7	73.7	6.2
Word	73.2	65.9	69.4	5.8
Char + Word	78.9	69.7	74.0	7.5

Table 4: Evaluating different input representations. The percentage of ill-formed DRSs is denoted by % ill.

word in the input and still initialize these embeddings using the pre-trained GloVe embeddings.

Representation results The results of the experiments (10-fold CV) for finding the best representation are shown in Table 4. Character representations are clearly better than word representations, though the word-level representation produces fewer ill-formed DRSs. Both representations are maintained for our further experiments. Although the combination of characters and words did lead to a small increase in performance over characters only (Table 4), this difference is not significant. Hence, this representation is discarded in further experiments described in this paper.

4.2 Tokenization

An interesting aspect of the PMB data is the way the input sentences are tokenized. In the data set, multiword expressions are tokenized as single words, for example, “New York” is tokenized to “New~York.” Unfortunately, most off-the-shelf tokenizers (e.g., the Moses tokenizer) are not equipped to deal with this. We experiment with using Elephant (Evang et al., 2013), a tokenizer that can be (re-)trained on individual data sets, using the tokenized sentences of the published silver and gold PMB data set.⁷ Simultaneously, we are interested in whether character-level models need tokenization at all, which would be a possible advantage of this type of representing the input text.

Results of the experiment are shown in Table 5. None of the two tokenization methods yielded a significant advantage for the character-level models, so they will not be used further. The word-level models, however, did benefit from tokenization, but Elephant did not give us an advantage over the Moses tokenizer. Therefore, for

⁶The Common Crawl version trained on 840 billion tokens, vector size 300.

⁷Gold tokenization is available in the data set, but using this would not reflect practical applications of DRS parsing, as we want raw text as input for a realistic setting.

b1 REF x1	\$1 REF @1	[NEW] REF <NEW>
b1 male "n.02" x1	\$1 male "n.02" @1	[0] male "n.02" <0>
b1 Name x1 "tom"	\$1 Name @1 "tom"	[0] Name <0> "tom"
b2 REF t1	\$2 REF @2	[NEW] REF <NEW>
b2 EQU t1 "now"	\$2 EQU @2 "now"	[0] EQU <0> "now"
b2 time "n.08" t1	\$2 time "n.08" @2	[0] time "n.08" <0>
b0 NOT b3	\$0 NOT \$3	[NEW] NOT [NEW]
b3 REF s1	\$3 REF @3	[0] REF <NEW>
b3 Time s1 t1	\$3 Time @3 @2	[0] Time <0> <-1>
b3 Experiencer s1 x1	\$3 Experiencer @3 @1	[0] Experiencer <0> <-2>
b3 afraid "a.01" s1	\$3 afraid "a.01" @3	[0] afraid "a.01" <0>
b3 Stimulus s1 x2	\$3 Stimulus @3 @4	[0] Stimulus <0> <1>
b3 REF x2	\$3 REF @4	[0] REF <NEW>
b3 entity "n.01" x2	\$3 entity "n.01" @4	[0] entity "n.01" <0>

(a) Standard naming

(b) Absolute naming

(c) Relative naming

Figure 4: Different methods of variable naming exemplified on the clausal form of Figure 1. For (c), positive numbers refer to introductions that have yet to occur, and negative numbers refer to known introductions. A zero refers to the previous introduction for that variable type.

word-level models, we use Moses in our subsequent experiments.

4.3 Representing Variables

So far we did not attempt to do anything special with the variables that occur in DRSs, as we simply tried to learn them as supplied in the PMB data set. Obviously, DRSs constitute a challenge for seq2seq models because of the high number of multiple occurrences of the same variables, in particular compared with AMR. AMR parsers do not deal well with this, because the reentrancy metric (Damonte et al., 2017) is among the lowest metrics for all AMR parsers that reported them or are publicly available (van Noord and Bos, 2017b). Moreover, for AMR, only 50% of the representations contain at least one reentrant node, and only 20% of the triples in AMR contain a reentrant node (van Noord and Bos, 2017a), but for DRSs these are both virtually 100%. Although seq2seq AMR parsers could get away with ignoring variables during training and reinstating them in a post-processing step, for DRSs this is unfeasible.

However, because variable names are chosen arbitrarily, they will be hard for a seq2seq model to learn. We will therefore experiment with two methods of rewriting the variables to a more general representation, distinguishing between box variables and discourse variables. Our first method (absolute) traverses down the list of clauses, rewriting each new variable to a unique representation, taking the order into account. The second

	Char parser		Word parser	
	F1	% ill	F1	% ill
Baseline (bs)	73.7	6.2	69.4	5.8
Moses (mos)	74.1	4.8	71.8	5.8
Elephant (ele)	74.0	5.4	71.1	7.5
bs/mos + absolute (abs)	75.3	3.5	73.5	2.0
bs/mos + relative (rel)	76.3	4.2	74.2	3.1
bs/mos + rel + lowercase	75.8	3.6	74.9	3.1
bs/mos + rel + truecase	76.2	4.0	73.3	3.3
bs/mos + rel + feature	76.9	3.7	74.9	2.9

Table 5: Results of the 10-fold CV experiments regarding tokenization, variable rewriting, and casing; bs/mos means that we use no tokenization for the character-level parser, while we use Moses for the word-level parser.

method (relative) is more sophisticated; it rewrites variables based on when they were introduced, inspired by the De Bruijn index (de Bruijn, 1972). We view box variables as introduced when they are first mentioned, and we take the REF clause of a discourse referent as their introduction. The two rewriting methods are illustrated in Figure 4.

The results are shown in Table 5. For both characters and words, the relative rewriting method significantly outperforms the absolute method and the baseline, though the absolute method produces fewer ill-formed DRSs. Interestingly, the character-level model still obtains a higher F1-score compared to the word-level model, even though it produces more ill-formed DRSs.

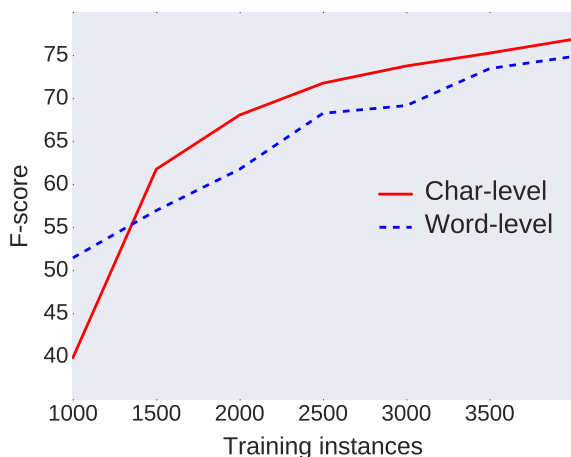


Figure 5: Learning curve for different number of gold instances for both the character-level and word-level neural parsers (10-fold CV experiment for every 500 instances).

4.4 Casing

Casing is a writing device mostly used for punctuation purposes. On the one hand, it increases the set of characters (hence adding more redundant variation to the input). On the other hand, case can be a useful feature to recognize proper names because names of individuals are semantically analysed as presuppositions. Explicitly encoding uppercase with a feature could therefore prevent us from including a named-entity recognizer, often used in other semantic parsers. Although we do not expect dealing with case to be a major challenge, we try out different techniques to find an optimal balance between abstracting over input characters and parsing performance. The results, in Table 5, show that the feature works well for the character-level model, but for the word-level model, it does not outperform lowercasing. These settings are used in further experiments.

5 Experiments with Additional Data

Because semantic annotation is a difficult and time-consuming task, gold standard data sets are usually relatively small. This means that semantic parsers (and data-hungry neural methods in particular) can often benefit from more training data. Some examples in semantic parsing are data recombination (Jia and Liang, 2016), paraphrasing (Berant and Liang, 2014), or exploiting machine-generated output (Konstas et al., 2017). However, before we do any experiments using extra training data, we want to be sure that we can still ben-

	Char parser		Word parser	
Data	F1	% ill	F1	% ill
Best gold-only	75.9	2.9	72.8	2.0
+ ensemble	77.9	1.8	75.1	0.9
Gold + silver	82.9	1.8	82.7	1.1
+ ensemble	83.6	1.3	83.1	0.7

Table 6: F1-score and percentage of ill-formed DRSs on the test set, for the experiments with the PMB-released silver data. The scores without using an ensemble are an average of five runs of the model.

efit from more gold training data. For both the character level and word level we plot the learning curve, adding 500 training instances at a time, in Figure 5. For both models the F-score clearly still improves when using more training instances, which shows that there is at least the potential for additional data to improve the score.

For DRSs, the PMB-2.1.0 release already contains a large set of silver standard data (71,308 instances), containing DRSs that are only partially manually corrected. We then train a model on both the gold and silver standard data, making no distinction between them during training. After training we take the last model and restart the training on only the gold data, in a similar process as described in Konstas et al. (2017) and van Noord and Bos (2017b). In general, restarting the training to fine-tune the weights of the model is a common technique in NMT (Denkowski and Neubig, 2017).

We are aware that there are many methods to obtain and utilize additional data. However, our main aim is not to find the optimal method for DRS parsing, but to demonstrate that using additional data is indeed beneficial for neural DRS parsing. Because we are not further fine-tuning our model, we will show results on the test set in this section.

Table 6 shows the results of adding the silver data. This results in a large increase in performance, for both the character- and word-level models. We are still reliant on manually annotated data, however, because without the gold data (so training on only the silver data), we score even lower than our baseline model (68.4 and 68.1 for the char and word parser). Similarly, we are reliant on the fine-tuning procedure, as we also score below our baseline models without it (71.6 and 71.0 for the char and word parsers, respectively).

Data	Char parser		Word parser	
	F1	% ill	F1	% ill
Silver (Boxer-generated)	83.6	1.3	83.1	0.7
Bronze (Boxer-generated)	83.8	1.1	82.4	0.9
Bronze (NN-generated)	77.9	2.7	74.5	2.2
without ill-formed DRSs	78.6	1.6	74.9	0.9

Table 7: Test set results of the experiments that analyze the impact of the silver data.

We believe there are two possible factors that could explain why the addition of silver data results in such a large improvement: (i) the fact that the data are silver instead of bronze or (ii) the fact that a different DRS parser (Boxer, see §6) is used to create the silver data instead of our own parser. We conduct an experiment to identify the impact on performance of silver versus bronze and Boxer versus our parser. The results are shown in Table 7. Note that these experiments are performed to analyze the impact of the silver data, not to further push the score, meaning *Silver (Boxer-generated)* is our final model that will be compared to other approaches in §6.

For factor (i), we compare the performance of the model trained on silver and bronze versions of the exact same documents (so leaving out the manual corrections). Interestingly, we score slightly higher for the character-level model with bronze than with silver (though the difference is not statistically significant), meaning that the extra manual corrections are not beneficial (in their current format). This suggests that the silver data are closer to bronze than to the gold standard.

For factor (ii), we use our own best parser (without silver data) to parse the sentences in the PMB silver data release and use that as additional training data.⁸ Because the silver data contain longer and more complicated sentences than the gold data, our best parser produces more ill-formed DRSs (13.7% for char and 15.6% for word). We can either discard those instances or still maintain them for the model to learn from. For Boxer this is not an issue since only 0.3% of DRSs produced were ill-formed. We observe that a full self-training pipeline results in lower performance compared with using Boxer-produced DRSs. In fact, this does not seem to be beneficial over only using the gold standard data. Most likely,

⁸Note that we cannot apply the manual corrections, so in PMB terminology, these data are bronze instead of silver.

	Prec	Rec	F-score
SPAR	48.0	33.9	39.7
SIM-SPAR	55.6	57.9	56.8
AMR2DRS	43.3	43.0	43.2
Boxer	75.7	72.9	74.3
Neural Char	79.7	76.2	77.9
Neural Word	77.1	73.3	75.1
Neural Char + silver	84.7	82.4	83.6
Neural Word + silver	84.0	82.3	83.1

Table 8: Test set results of our best neural models compared to two baseline models and two parsers.

because Boxer combines symbolic and statistical methods, it learns very different things from our neural parsers, which in turn provides more valuable information to the model. A more detailed analysis on the difference in (semantic) output is performed in §6.2 and 6.3. Removing ill-formed DRSs before training leads to higher F-scores for both the char and word parsers, as well as a lower number of ill-formed DRSs.

6 Discussion

6.1 Comparison

In this section, we compare our best neural models (with and without silver data, see Table 6) with two baseline systems and with two DRS parsers: AMR2DRS and Boxer. AMR2DRS is a parser that obtains DRSs from AMRs by applying a set of rules (van Noord et al., 2018), in our case using AMRs produced by the AMR parser of van Noord and Bos (2017b). Boxer is an existing DRS parser using a statistical combinatorial categorical grammar parser for syntactic analysis and a compositional semantics based on λ -calculus, followed by pronoun and presupposition resolution (Curran et al., 2007; Bos, 2008b). SPAR is a baseline parser that outputs the same (fixed) default DRS for each input sentence. We implemented a second baseline model, SIM-SPAR, which outputs, for each sentence in the test set, the DRS of the most similar sentence in the training set. This similarity is calculated by taking the cosine similarity of the average word embedding vector (with removed stopwords) based on the GloVe embeddings (Pennington et al., 2014).

Table 8 shows the result of the comparison. The neural models comfortably outperform the baselines. We see that both our neural models

	Char	Word	Boxer
All clauses	83.6	83.1	74.3
DRS Operators	93.2	93.3	88.0
VerbNet roles	84.1	82.5	71.4
WordNet synsets	79.7	79.4	72.5
nouns	86.1	88.5	82.5
verbs, adverbs, adj.	65.1	58.7	49.3
Oracle sense numbers	86.7	85.7	78.1
Oracle synsets	90.7	90.9	83.8
Oracle roles	87.4	87.2	82.0

Table 9: F-scores of fine-grained evaluation on the test set of the three semantic parsers.

outperform Boxer by a large margin when using the Boxer-labeled silver data. However, even without this dependence, the neural models perform significantly better than Boxer. It is worth noting that the character-level model significantly outperforms the word-level model, even though it cannot benefit from pre-trained word embeddings and from a tokenizer.

Concurrently with our work, a neural DRS parser has been developed by Liu et al. (2018). They use a customized neural seq2seq model that produces the DRS in three stages. It first predicts the general (deep) structure of the DRSs, after which the conditions and referents are filled in. Unfortunately, they train and evaluate their parser on annotated data from the GMB rather than from the PMB (see §2). This, combined with the fact that their work is contemporaneous to the current paper, make it difficult to compare the approaches. However, we see no apparent reason why their method should not work on the PMB data.

6.2 Analysis

An intriguing question is what our models actually learn, and what parts of meaning are still challenging for neural methods. We do this in two ways, by performing an automatic analysis and by doing a manual inspection on a variety of semantic phenomena. Table 9 shows an overview of the different automatic evaluation metrics we implemented, with corresponding scores of the three models.

The character- and word-level systems perform comparably in all categories except for VerbNet roles, where the character-based parser shows a clear advantage (1.6 percentage point difference). The score for WordNet synsets is similar, but

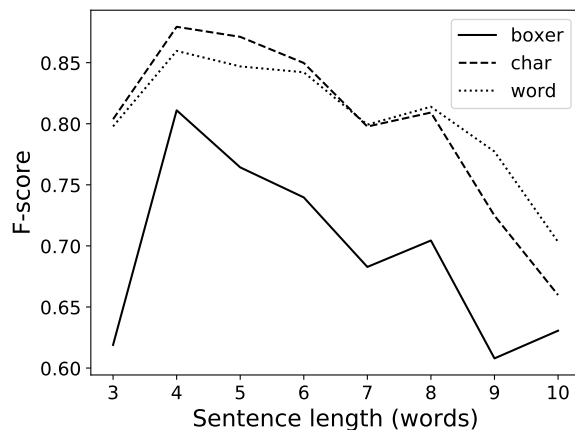


Figure 6: Performance of each parser for sentences of different length.

the word-level model has more difficulty predicting synsets that are introduced by verbs than for nouns. It is clear that the neural models outperform Boxer consistently on each of these metrics (partly because Boxer picks the first sense by default). What also stands out is the impact of the word senses: With a perfect word sense-disambiguation module (oracle senses), large improvements can be gained for all three parsers.

It is interesting to look at what errors the model makes in terms of producing ill-formed output. For both the neural parsers, only about 2% of the ill-formed DRSs are ill-formed because of a syntactic error in an individual clause (e.g., `b1 Agent x1`, where a fourth argument is missing), whereas all the other errors are due to a violated semantic constraint (see §3.2). In other words, the produced output is a syntactically well-formed DRS but is not interpretable.

To find out how sentence length affects performance, we plot in Figure 6 the mean F-score obtained by each parser on input sentences of different lengths, from 3 to 10 words.⁹ We observe that all the parsers degrade with sentence length. To identify whether any of the parsers degrades significantly more than any other, we build a regression model in which we predict the F-score using as predictors the parser (char, word, and Boxer), the sentence length, and the number of clauses produced. According to the regression model, (i) the performance of all three

⁹Shorter and longer sentences are excluded as there are fewer than 10 input sentences for any such length—for example, there are only three sentences that have two words.

systems decreases with sentence length, thus corroborating the trends shown in Figure 6 and (ii) the interaction between parser and sentence length is not significant (i.e., none of the parsers decreases significantly more than any other with sentence length). The fact that the performance of the neural parsers degrades with sentence length is not surprising, because they are based on the seq2seq architecture, and models built on this architecture for other tasks, such as machine translation, have been shown to have the same issue (Torralba and Sánchez-Cartagena, 2017).

6.3 Manual Inspection

The automatic evaluation metrics provide overall scores but do not capture how the models perform on certain semantic phenomena present in the DRSs. Therefore, we manually inspected the test set output of the three parsers for the semantic phenomena listed in Table 2. We here describe each phenomenon and explain how the parser output is evaluated on them.

The **negation & modals** phenomenon covers possibility (POS), necessity (NEC), and negation (NOT). The phenomenon is considered successfully captured if an automatically produced clausal form has the clause with the modal operator and the main concept is correctly put under the scope of the modal operator. For example, to capture the negation in Figure 1, the presence of **b0** NOT **b3** and **b3** afraid "a.01" **s1** is sufficient. **Scope ambiguity** counts nested pairs of scopal operators such as possibility (POS), necessity (NEC), negation (NOT), and implication (IMP). **Pronoun resolution** checks if an anaphoric pronoun and its antecedent are represented by the same discourse referent. **Discourse relation & implication** involves determining a discourse relation or an implication with a main concept in each of their scopes (i.e., boxes). For instance, to get the discourse relation in Figure 2 correctly, a clausal form needs to include **b0** CONTINUATION **b1** **b5**, **b1** play "v.03" **e1**, and **b5** sing "v.01" **e2**. Finally, the **embedded clauses** phenomenon verifies whether the main verb concept of an embedded clause is placed inside the propositional box (PRP). This phenomenon also covers control verbs: It checks whether a controlled argument of a subordinate verb is correctly identified as an argument of a control verb.

Phenomenon	#	Char	Word	Boxer
Negation & modals	73	0.90	0.81	0.89
Scope ambiguity	15	0.73	0.57	0.80
Pronoun resolution	31	0.84	0.77	0.90
Discourse rel. & imp.	33	0.64	0.67	0.82
Embedded clauses	30	0.77	0.70	0.87

Table 10: Manual evaluation of the output of the three semantic parsers on several semantic phenomena. Reported numbers are accuracies.

The results of the semantic evaluation of the parsers on the test set is given in Table 10. The character-level parser performs better than the word-level parser on all the phenomena except one. Even though both our neural parsers clearly outperformed Boxer in terms of F-score, they perform worse than Boxer on the selected semantic phenomena. Although the differences are not big, Boxer obtained the highest score for four out of five phenomena. This suggests that just the F-score is perhaps not good enough as an evaluation metric, or that the final F-score should perhaps be weighted towards certain clauses. For example, it is arguably more important to capture a negation correctly than tense. Our current metric only gives a rough indication about the contents, but not about the inferential capabilities of the meaning representation.

7 Conclusions and Future Work

We implemented a general, end-to-end neural seq2seq model that is able to produce well-formed DRSs with high accuracy (RQ1). Character-level models can outperform word-level models, even though they are not dependent on tokenization and pre-trained word embeddings (RQ2). It is beneficial to rewrite DRS variables to a more general representation (RQ3). Obtaining and using additional data can benefit performance as well, though it might be better to use an external parser rather than doing a full self-training pipeline (RQ4). F-score is only a rough measure for semantic accuracy: Boxer still outperformed our best neural models on a subset of specific semantic phenomena (RQ5).

We think there are many opportunities for future work. Because the sentences in the PMB data set are relatively short, it makes sense to investigate seq2seq models performing well for longer texts. There are a few promising directions here

that could combat the degrading performance on longer sentences. First, the Transformer model (Vaswani et al., 2017) is an interesting candidate for exploration, a state-of-the-art neural model developed for MT that does not have worse performance for longer sentences. Second, a seq2seq model that is able to first predict the general structure of the DRS, after which it can fill in the details, similar to Liu et al. (2018), is something that could be explored. A third possibility is a neural parser that tries to build the DRS incrementally, producing clauses for different parts of the sentence individually, and then combining them to a final DRS.

Concerning the evaluation of DRS parsers, we feel there are a couple of issues that could be addressed in future work. One idea is to facilitate computing F-scores tailored to specific semantic phenomena that are dubbed important, so the evaluation we performed in this paper manually could be carried out automatically. Another idea is to evaluate the application of DRSs to improve performance on other linguistic or semantic tasks in which DRSs that capture the full semantics will, presumably, have an advantage. A combination of glass-box and black-box evaluation seems a promising direction here (Bos, 2008a; van Noord et al., 2018).

Acknowledgments

This work was funded by the NWO-VICI grant “Lost in Translation—Found in Meaning” (288-89-003). The Tesla K40 GPU used in this work was kindly donated to us by the NVIDIA Corporation. We also want to thank the three anonymous reviewers for their comments.

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 and Johan Bos. 2017. [Towards universal semantic tagging](#). In *Proceedings of the 12th International Conference on Computational Semantics (IWCS 2017) – Short Papers*, Montpellier, France. Association for Computational Linguistics.

David Alvarez-Melis and Tommi S. Jaakkola. 2017. Tree-structured decoding with doubly-recurrent neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Nicholas Asher. 1993. *Reference to Abstract Objects in Discourse*. Kluwer Academic Publishers.

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Studies in natural language processing. Cambridge University Press.

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 semantic banking](#). In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria.

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3196–3200, Istanbul, Turkey.

David I. Beaver. 2002. Presupposition projection in DRT: A critical assesment. In *The Construction of Meaning*, pages 23–43. Stanford University.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1415–1425.

Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3531–3541, Osaka, Japan.

- Patrick Blackburn and Johan Bos. 2005. *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. [Findings of the 2017 conference on machine translation \(WMT17\)](#). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.
- Claire Bonial, William J. Corvey, Martha Palmer, Volha Petukhova, and Harry Bunt. 2011. A hierarchical unification of LIRICS and VerbNet semantic roles. In *Proceedings of the 5th IEEE International Conference on Semantic Computing (ICSC 2011)*, pages 483–489.
- Johan Bos. 2008a. Let’s not argue about semantics. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008)*, pages 2835–2840, Marrakech, Morocco.
- Johan Bos. 2008b. [Wide-coverage semantic analysis with boxer](#). In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, volume 1 of *Research in Computational Semantics*, pages 277–286. College Publications.
- Johan Bos. 2015. Open-domain semantic parsing with Boxer. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 301–304.
- 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*. Springer Netherlands.
- Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1451.
- Nicolaas Govert de Bruijn. 1972. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the church-rosser theorem. In *Indagationes Mathematicae (Proceedings)*, volume 75, pages 381–392. Elsevier.
- Jan Buys and Phil Blunsom. 2017. Robust incremental neural semantic graph parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1215–1226.
- 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.
- James Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 33–36, Prague, Czech Republic.
- Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for abstract meaning representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain. Association for Computational Linguistics.
- Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 18–27, Vancouver. Association for Computational Linguistics.
- Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.
- Jan van Eijck and Hans Kamp. 1997. Representing discourse in context. In Johan van

- Bentham and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 179–240. Elsevier, MIT.
- Kilian Evang, Valerio Basile, Grzegorz Chrupała, and Johan Bos. 2013. **Elephant: Sequence labeling for word and sentence segmentation**. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1422–1426, Seattle, Washington, USA.
- Christiane Fellbaum, editor. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press, Cambridge, Ma., USA.
- Bart Geurts. 1999. *Presuppositions and Pronouns*, volume 3 of *Current Research in the Semantic-/Pragmatics interface*. Elsevier.
- Xiaodong He and David Golub. 2016. Character-level question answering with attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1598–1607.
- Benjamin Heinzerling and Michael Strube. 2018. BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).
- Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 12–22.
- Mark Johnson and Ewan Klein. 1986. Discourse, anaphora and parsing. In *11th International Conference on Computational Linguistics. Proceedings of Coling '86*, pages 669–675, University of Bonn.
- Nirit Kadmon. 2001. *Formal Pragmatics*. Blackwell.
- Hans Kamp. 1984. A theory of truth and semantic representation. In Jeroen Groenendijk, Theo M.V. Janssen, and Martin Stokhof, editors, *Truth, Interpretation and Information*, pages 1–41. FORIS, Dordrecht – Holland/Cinnaminson – U.S.A.
- Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. **Open-NMT: Open-source toolkit for neural machine translation**. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.
- Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. **Neural AMR: Sequence-to-sequence models for parsing and generation**. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada. Association for Computational Linguistics.
- Phong Le and Willem Zuidema. 2012. Learning compositional semantics for open domain semantic parsing. *Proceedings of COLING 2012*, pages 1535–1552.
- Wang Ling, Phil Blunsom, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Fumin Wang, and Andrew Senior. 2016. Latent predictor networks for code generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 599–609.
- Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2018. Discourse representation structure parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 429–439.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.
- Reinhard Muskens. 1996. Combining montague semantics and discourse representation. *Linguistics and Philosophy*, 19:143–186.

- 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)*, Paris, France. European Language Resources Association (ELRA).
- Rik van Noord and Johan Bos. 2017a. Dealing with co-reference in neural semantic parsing. In *Proceedings of the 2nd Workshop on Semantic Deep Learning (SemDeep-2)*, pages 41–49.
- Rik van Noord and Johan Bos. 2017b. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal*, 7:93–108.
- Eric W. Noreen. 1989. *Computer-intensive Methods for Testing Hypotheses*. Wiley New York.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Fernando Pereira and Stuart Shieber. 1987. *Prolog and Natural Language Analysis*. CSLI Lecture Notes 10. Chicago University Press, Stanford.
- Rob A. Van der Sandt. 1992. [Presupposition projection as anaphora resolution](#). *Journal of Semantics*, 9(4):333–377.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 371–376.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.
- Antonio Toral and Víctor M. Sánchez-Cartagena. 2017. [A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1063–1073, Valencia, Spain. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.
- Hajime Wada and Nicholas Asher. 1986. BUILDERS: An implementation of DR theory and LFG. In *11th International Conference on Computational Linguistics. Proceedings of Coling '86*, pages 540–545, University of Bonn.
- John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055.