Character-level Representations Improve DRS-based Semantic Parsing Even in the Age of BERT

van Noord, Rik; Toral Ruiz, Antonio; Bos, Johan

Published in:
Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)

DOI:
10.18653/v1/2020.emnlp-main.371

Character-level Representations Improve DRS-based Semantic Parsing Even in the Age of BERT

Rik van Noord
CLCG
University of Groningen
The Netherlands
rikvannoord@gmail.com

Antonio Toral
CLCG
University of Groningen
The Netherlands
a.toral.ruiz@rug.nl

Johan Bos
CLCG
University of Groningen
The Netherlands
johan.bos@rug.nl

Abstract

We combine character-level and contextual language model representations to improve performance on Discourse Representation Structure parsing. Character representations can easily be added in a sequence-to-sequence model in either one encoder or as a fully separate encoder, with improvements that are robust to different language models, languages and data sets. For English, these improvements are larger than adding individual sources of linguistic information or adding non-contextual embeddings. A new method of analysis based on semantic tags demonstrates that the character-level representations improve performance across a subset of selected semantic phenomena.

1 Introduction

Character-level models have obtained impressive performance on a number of NLP tasks, ranging from the classic POS-tagging (Santos and Zadrozny, 2014) to complex tasks such as Discourse Representation Structure (DRS) parsing (van Noord et al., 2018b). However, this was before the large pretrained language models (Peters et al., 2018; Devlin et al., 2019) took over the field, with the consequence that for most NLP tasks, state-of-the-art performance is now obtained by fine-tuning on one of these models (e.g., Conneau et al., 2020).

Does this mean that, despite a long tradition of being used in language-related tasks (see Section 2.1), character-level representations are no longer useful? We try to answer this question by looking at semantic parsing, specifically DRS parsing (Abzianidze et al., 2017; van Noord et al., 2018a). We aim to answer the following research questions:

1. Do pretrained language models (LMs) outperform character-level models for DRS parsing?
2. Can character and LM representations be combined to improve performance, and if so, what is the best method of combining them?
3. How do these improvements compare to adding linguistic features?
4. Are the improvements robust across different pretrained language models, languages, and data sets?
5. On what type of sentences do character-level representations specifically help?

Why semantic parsing? Semantic parsing is the task of automatically mapping natural language utterances to interpretable meaning representations. The produced meaning representations can then potentially be used to improve downstream NLP applications (e.g., Issa et al., 2018; Song et al., 2019; Mihaylov and Frank, 2019), though the introduction of large pretrained language models has shown that explicit formal meaning representations might not be a necessary component to achieve high accuracy. However, it is now known that these models lack reasoning capabilities, often simply exploiting statistical artifacts in the data sets, instead of actually *understanding* language (Niven and Kao, 2019; McCoy et al., 2019). Moreover, Ettinger (2020) found that the popular BERT model (Devlin et al., 2019) completely failed to acquire a general understanding of negation. Related, Bender and Koller (2020) contend that meaning cannot be learned from form alone, and argue for approaches that focus on grounding the language (communication) in the real world. We believe formal meaning representations therefore have an important role to play in future semantic applications, as semantic parsers produce an explicit model of a real-world interpretation.

Why Discourse Representation Structures? DRS parsing is a task that combines logical, pragmatic and lexical components of semantics in a

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 4587–4603,
November 16–20, 2020.

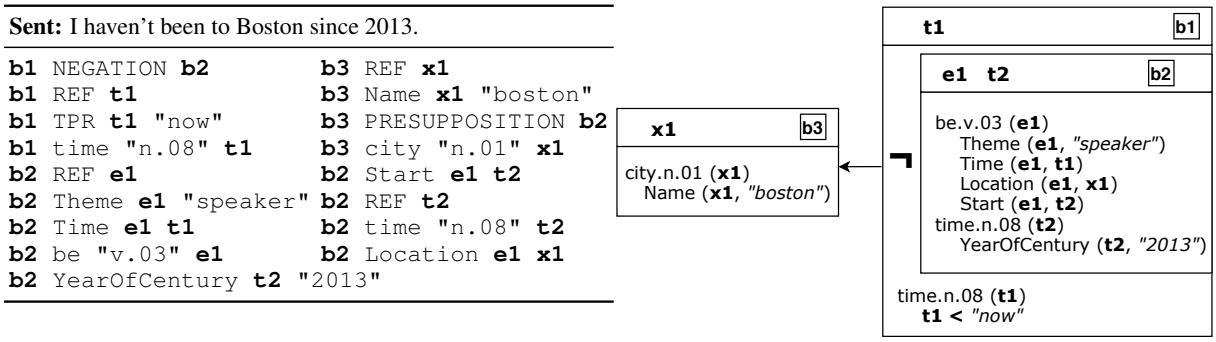


Figure 1: Example DRS in both clause (left) and box (right) representation.

single meaning representation. The task is complex and comprises other NLP tasks, such as semantic role labeling, word sense disambiguation, co-reference resolution and named entity tagging. Also, DRSs show explicit scope for certain operators, which allows for a more principled and linguistically motivated treatment of negation, modals and quantification, as has been advocated in formal semantics. Moreover, DRSs can be translated to formal logic, which allows for automatic forms of inference by third parties. Lastly, annotated DRSs are available in four languages (Abzianidze et al., 2017, see Section 3.3), allowing us to evaluate our models on multiple languages.

2 Background

2.1 Character-level models

The power of character-level representations has long been known in the field. In earlier work, they were successfully used in a range of tasks, including text-to-speech (Sejnowski and Rosenberg, 1987), parallel text alignment (Church, 1993), grapheme to phoneme conversion (Kaplan and Kay, 1994), language identification (Dunning, 1994), topical similarity prediction (Cavnar, 1994), named entity recognition (Klein et al., 2003), authorship attribution (Peng et al., 2003) and statistical machine translation (Vilar et al., 2007).

More recently, they also proved useful as input representations for neural networks, starting with success in general language modelling (Sutskever et al., 2011; Kim et al., 2016; Bojanowski et al., 2017), but also for a range of other tasks, including tokenization (Evang et al., 2013), POS-tagging (Santos and Zadrozny, 2014; Plank et al., 2016), dependency parsing (Ballesteros et al., 2015; Vania et al., 2018) and neural machine translation (Chung et al., 2016; Costa-jussà and Fonollosa, 2016; Luong and Manning, 2016; Cherry et al., 2018).

In semantic parsing, if character-level representations are employed, they are commonly used in combination with non-contextual word-level representations (Lewis et al., 2016; Ballesteros and Al-Onaizan, 2017; Groschwitz et al., 2018; Cai and Lam, 2019). There are a few recent studies that did use character-level representations in combination with BERT (Zhang et al., 2019a,b; Cai and Lam, 2020), though only Zhang et al. (2019a) provided an ablation score without the characters. Moreover, it is not clear if this small improvement was significant. van Noord and Bos (2017) and van Noord et al. (2018b), on the other hand, used solely character-level representations in an end-to-end fashion, using a bi-LSTM sequence-to-sequence model, which outperformed word-based models that employed non-contextual embeddings.

2.2 Discourse Representation Structures

DRSs are formal meaning representations introduced by Discourse Representation Theory (Kamp and Reyle, 1993) with the aim to capture the meaning of texts (Figure 1). Many variants of DRS have been proposed throughout the years. We adopt Venhuizen et al. (2018)'s version of DRT, which is close to Kamp's original ideas, but has a neo-Davidsonian view of event semantics and explicitly represents presuppositions.

Corpora The Groningen Meaning Bank (GMB, Basile et al., 2012; Bos et al., 2017) was the first attempt of annotating open domain English texts with DRSs. The released documents are partially corrected, but there are no gold standard sets available for evaluation. A similar corpus is the Parallel Meaning Bank (PMB, Abzianidze et al., 2017), which builds upon the GMB in a number of ways. It contains (parallel) texts in four languages: English, German, Italian and Dutch, with more fine-grained and language-neutral DRSs. Semantic tags are used during annotation (Bjerva et al., 2016; Abzianidze and Bos, 2017), and all non-logical DRS symbols

are grounded in either WordNet (Fellbaum, 1998) or VerbNet (Bonial et al., 2011). Moreover, its releases contain gold standard DRSs. For these reasons, we take the PMB as our corpus of choice to evaluate our DRS parsers.

DRS parsing Early approaches to DRS parsing employed rule-based systems for small English texts (Johnson and Klein, 1986; Wada and Asher, 1986; Bos, 2001). The first open domain DRS parser is Boxer (Bos, 2008, 2015), which is a combination of rule-based and statistical models. Le and Zuidema (2012) used a probabilistic parsing model that used dependency structures to parse GMB data as graphs. More recently, Liu et al. (2018) proposed a neural model that produces (tree-structured) DRSs in three steps by first learning the general (box) structure of a DRS, after which specific conditions and referents are filled in. In follow-up work (Liu et al., 2019a) they extend this work by adding an improved attention mechanism and constraining the decoder to ensure well-formed output. This model achieved impressive performance on both sentence-level and document-level DRS parsing on GMB data. Fu et al. (2020) in turn improve on this work by employing a Graph Attention Network during both encoding and decoding.

The introduction of gold standard DRSs in the PMB enabled a principled comparison of approaches. In our previous work (van Noord et al., 2018b), we showed that sequence-to-sequence models can successfully learn to produce DRSs, with characters as the preferred representation. In follow-up work, we improved on these scores by adding linguistic features (van Noord et al., 2019). The first shared task on DRS parsing (Abzianidze et al., 2019) sparked more interested in the topic, with a system based on stack-LSTMs (Evang, 2019) and a neural graph-based system (Fancellu et al., 2019). The best system (Liu et al., 2019b) used a similar approach as van Noord et al. (2018b), but swapped the bi-LSTM encoder for a Transformer. We will compare our approach to these models in Section 4.

3 Method

3.1 Neural Architecture

As our baseline system, we start from a fairly standard sequence-to-sequence model with attention (Bahdanau et al., 2015), implemented in AllenNLP (Gardner et al., 2017).[1] We improve on this model in a number of ways, mainly based on Nematus (Sennrich et al., 2017): (i) we initialize the decoder hidden state with the mean of all encoder states, (ii) we add an extra linear layer between this mean encoder state and the initial decoder state and (iii) we add an extra linear layer after each decoder state.

Specifically, given a source sequence $(s_1, \ldots, s_l)$ of length l, and a target sequence $(t_1, \ldots, t_k)$ of length k, let $\mathbf{e}_i$ be the embedding of source symbol i, let $\mathbf{h}_i$ be the encoder hidden state at source position i and let $\mathbf{d}_j$ be the decoder state at target position j. A single forward encoder state is obtained as follows: $\overrightarrow{\mathbf{h}}_i = \text{LSTM}(\overrightarrow{\mathbf{h}}_{i-1}, \mathbf{e}_i)$. The final state is obtained by concatenating the forward and backward hidden states, $\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i]$. The decoder is initialized with the average over all encoder states: $\mathbf{c}_{\text{tok}} = \left(\sum_{i=1}^{l} \mathbf{h}_i\right) / \, l$ and $\mathbf{d}_0 = \tanh\left(\mathbf{W}_{\text{init}}\, \mathbf{c}_{\text{tok}}\right)$.

Characters in one encoder We will experiment with adding character-level information in either one or two encoders. For one encoder, we use char-CNN (Kim et al., 2016), which runs a Convolutional Neural Network (LeCun et al., 1990) over the characters for each token. It applies convolution layers for certain *widths*, which in essence select n-grams of characters. For each width, it does this a predefined number of times, referred to as the number of *filters*. The filter vectors form a matrix, which is then pooled to a vector by taking the max value of each initial filter vector. A detailed schematic overview of this procedure is shown in Appendix A. However, we usually do not look at only a single width, but at a range of widths, e.g., $[1, 2, 3, 4, 5]$. In that case, we simply concatenate the resulting vectors to obtain our final char-CNN embedding: $\mathbf{e}_{\text{char}_i} = [\mathbf{e}_{w1}; \mathbf{e}_{w2}; \mathbf{e}_{w3}; \mathbf{e}_{w4}; \mathbf{e}_{w5}]$. Each width-filter combination has independent learnable parameters. Finally, the char-CNN embedding is concatenated to the token-level representation, which is fed to the encoder: $\mathbf{e}_i = [\mathbf{e}_{\text{tok}_i}; \mathbf{e}_{\text{char}_i}]$.

Characters in two encoders In the two-encoder setup, we run separate (but structurally identical) bi-LSTM encoders over the tokens and characters, and concatenate the resulting context vector before we feed it to the decoder:

$$\mathbf{d}_0 = \tanh\left(\mathbf{W}_{\text{init}}\, [\mathbf{c}_{\text{tok}}; \mathbf{c}_{\text{char}}]\right)$$

In the decoder, we replace the LSTM with a doubly-attentive LSTM, based on the doubly-

[1] https://github.com/RikVN/allennlp

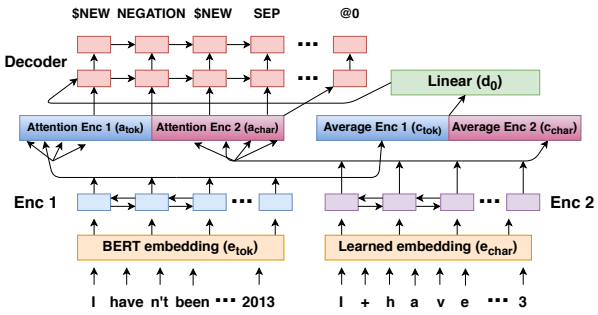


Figure 2: Schematic overview of our neural architecture when using two encoders (BERT and characters).

attentive GRU (Calixto et al., 2017). We apply soft-dual attention (Junczys-Dowmunt and Grundkiewicz, 2017) to be able to attend over both encoders in the decoder (also see Figure 2):

$$\mathbf{d}'_j = \text{LSTM}_1\left(\mathbf{d}_{j-1}, \mathbf{e}_{t_{j-1}}\right)$$
$$\mathbf{a}_j = \left[\text{ATT}\left(\mathbf{C}_{\text{tok}}, \mathbf{d}'_j\right); \text{ATT}\left(\mathbf{C}_{\text{char}}, \mathbf{d}'_j\right)\right]$$
$$\mathbf{d}_j = \text{LSTM}_2\left(\mathbf{d}'_j, \mathbf{a}_j\right)$$

Here, $\mathbf{e}_{t_{j-1}}$ is the embedding of the previously decoded symbol t, C the set of encoder hidden states for either the tokens or characters, ATT the attention function (dot-product) and $\mathbf{d}_j$ the final decoder hidden state at step j. This model can easily be extended to more than two encoders, which we will experiment with in Section 4.

This type of multi-source model is commonly used to represent different languages, e.g., in machine translation (Zoph and Knight, 2016; Firat et al., 2016) and semantic parsing (Susanto and Lu, 2017; Duong et al., 2017), though it has also been successfully applied in multi-modal translation (Libovický and Helcl, 2017), multi-framework semantic parsing (Stanovsky and Dagan, 2018) and adding linguistic information (Currey and Heafield, 2018; van Noord et al., 2019). To the best of our knowledge, we are the first to represent the characters as a source of extra information in a multi-source sequence-to-sequence model.

Transformer We also experiment with the Transformer model (Vaswani et al., 2017), using the stacked self attention model as implemented in AllenNLP. A possible advantage of this model is that it might handle longer sentences and documents better. However, it might be harder to tune (Popel and Bojar, 2018)[2] and its improved performance has mainly been shown for large data sets, as opposed to the generally smaller semantic parsing data sets (Section 3.3). Indeed, we cannot outperform the LSTM architecture (see Section 4), even when tuning more extensively. We therefore do not experiment with adding character-level representations to this architecture, though the char-CNN can be added similarly as for the LSTM model.

Hyper-parameters To make a fair comparison, we conduct an independent hyper-parameter search on the development set for all nine input text representations (see Section 3.2) across the two neural architectures, starting from the settings of van Noord et al. (2019). We found that the best settings were very close for all systems, with the only notable difference that the learning rate of the Transformer models is considerably smaller than for the bi-LSTM models (0.0002 vs 0.001).[3]

For the char-CNN model, we use 100 filters, an embedding size of 75 and n-gram filter sizes of $[1, 2, 3]$ for English and $[1, 2, 3, 4, 5]$ for German, Italian and Dutch. For experiments where we add characters or linguistic features, the only extra search we do is the size of the hidden vector of the RNN encoder ($300 - 600$), since this vector now has to contain more information, and could potentially benefit from a larger size. Note that (possible) improved performance is not simply due to larger model capacity, since during tuning of the baseline models a larger RNN hidden size did not result in better performance.

3.2 Representations

We will experiment with five well-known pre-trained language models: ELMO (Peters et al., 2018), BERT base/large (Devlin et al., 2019) and ROBERTA base/large (Liu et al., 2019c).[4] The performance of these five large LMs is contrasted with results of a character-level model and three word-based models. The word-based models either learn the embeddings from scratch or use non-contextual GLOVE (Pennington et al., 2014) or FASTTEXT (Grave et al., 2018) embeddings. Pre- and postprocessing of the DRSs is done using the method described in van Noord et al. (2018b).[5] The DRSs are linearized, after which the variables are rewritten to a relative representation. The character-level model

[2] Also see: https://twitter.com/srush_nlp/status/1245825437240102913

[3] See Appendix B for specific hyperparameter settings.

[4] We are aware that there exist several other large pre-trained language models (e.g., Yang et al., 2019; Raffel et al., 2020; Clark et al., 2020), but we believe that the models we used have had the largest impact on the field.

[5] https://github.com/RikVN/Neural_DRS/

		Gold			Silver	Bronze
		Train	Dev	Test	Train	Train
2.2.0	English	4,597	682	650	67,965	120,662
	German	0	727	747	4,235	102,998
	Italian	0	374	400	2,515	61,504
	Dutch	0	370	341	1,051	20,554
3.0.0	English	6,620	885	898	97,598	146,371
	German	1,159	417	403	5,250	121,111
	Italian	0	515	547	2,772	64,305
	Dutch	0	529	483	1,301	21,550

Table 1: Number of documents for the four languages, for the two PMB releases considered.

has character representations for the DRS concepts and constants, but not for variables, roles and operators. For all word-level models, the DRS concepts are initialized with GLOVE embeddings, while the other target tokens are learned from scratch.

BERT specifics For the BERT models, we obtained the best performance by only keeping the vector of the first WordPiece per original token (e.g., only keep `play` out of `play ##ing`). For ROBERTA, it was best to use the WordPiece tokenization as is. Since linguistic features are added on token level, we duplicate the semantic tags for multi-piece tokens of ROBERTA in Table 5. Interestingly, we found that for both BERT and ROBERTA, it was best to keep the pretrained weights frozen. This was not a small difference: models using fine-tuning always obtained low scores (45 to 60).

3.3 Data and Evaluation

We use PMB releases 2.2.0 and 3.0.0[6] in our experiments (Table 1). The latter is a larger and more diverse extension of 2.2.0, which will be used for most of our experiments. We use 2.2.0 to compare to previous work and to verify that our results are robust across datasets. The PMB releases contain DRSs for four languages (English, German, Italian and Dutch) for three levels of annotation: gold (fully manually checked), silver (partially manually corrected) and bronze (no manual corrections). To make a fair comparison to previous work, we only employ the gold and silver data, by pretraining on gold + silver data and subsequently fine-tuning on only the gold data. If there is no gold train data available, we train on silver + bronze and fine-tune on silver. Unless otherwise indicated, our results are on the English development set of release 3.0.0.

[6] https://pmb.let.rug.nl/data.php

Sent	I	have	n't	been	to	Boston	since	2013
POS	PRP	VBP	RB	VBN	TO	NNP	IN	CD
SEM	PRO	NOW	NOT	EXT	REL	GPE	REL	YOC
LEM	I	have	not	be	to	Boston	since	2013
DEP	nsubj	aux	neg	cop	case	ROOT	case	nmod
CCG	NP	VP\VP	VPVP	VP/PP	PP/NP	N	(VP\VP)/NP	N

Table 2: Example representation for each source of linguistic information (PMB document p00/d1489).

Linguistic features We want to contrast our method of character-level information to adding sources of linguistic information. Based on van Noord et al. (2019), we employ these five sources: part-of-speech tags (POS), dependency parses (DEP), lemmas (LEM), CCG supertags (CCG) and semantic tags (SEM). For the first three sources, we use Stanford CoreNLP (Manning et al., 2014) to parse the documents in our dataset. The CCG supertags are obtained by using easyCCG (Lewis and Steedman, 2014). For semantic tagging, we train our own trigram-based tagger using TnT (Brants, 2000).[7] Table 2 shows a tagged example sentence for all five sources of information. Moreover, we also include non-contextual GLOVE and FASTTEXT embeddings as an extra source of information.

We add these sources of linguistic information in the same way as we add the character-level information, in either one or two encoders (see Section 3.1). In two encoders, we can use the exact same architecture. For one encoder, we (obviously) do not use the char-CNN, but learn a separate embedding for the tags (of size 200), that is then concatenated to the token-level representation, i.e., $\mathbf{e}_i = [\mathbf{e}_{\text{tok}_i}; \mathbf{e}_{\text{ling}_i}]$. If we use two encoders with a LM, characters *and* linguistic information (e.g., Table 4), the characters are added separately in the second encoder, while the LM and linguistic information representations are added in the first encoder.

Evaluation We compare the produced DRSs to the gold standard using Counter (van Noord et al., 2018a), which calculates micro precision, recall and F1-score based on the number of matching clauses.[8] We use Referee (van Noord et al., 2018b) to ensure that the produced DRSs are syntactically and semantically well-formed (i.e., no free variables, no loops in subordinate relations) and form a connected graph. DRSs that are ill-formed get

[7] This tagger is also used in the PMB pipeline, see Abzianidze and Bos (2017). It outperformed an ngram-based CRF-tagger (Lafferty et al., 2001) we also tried, obtaining an accuracy of 94.4% on the dev set.

[8] https://github.com/RikVN/DRS_parsing/

an F1-score of 0.0. All shown scores are averaged F1-scores over five training runs of the system, in which the same five random seeds are used.[9] For significance testing we use approximate randomization (Noreen, 1989), with $\alpha = 0.05$ and $R = 1000$.

We also introduce and release **DRS-JURY**. This program provides a detailed overview of the performance of a DRS parser, but can also compare experiments, possibly over multiple runs. Features include significance testing, semantic tag analysis (Section 5.1), sentence length plotting (Section 5.2), new detailed Counter scores (Appendix D), and analysing (relative) best/worst produced DRSs (Appendix E). We hope this is a step in the direction of a more principled way of evaluating DRS parsers.

4 Results

LMs vs char-level models DRS parsing is no exception to the general trend in NLP: it is indeed the case that the pretrained language models outperform the char-only model (Table 3). Interestingly, the Transformer model has worse performance for all representations.[10] Surprisingly, we find that BERT-BASE is the best model, though the differences are small.[11] We use this model in further experiments (referred to as BERT).

Adding characters to BERT We can see the impact of adding characters to BERT (first row of results in Table 4). For both methods, it results in a clear and significant improvement over the BERT-only baseline, 87.6 versus 88.1.

Adding linguistic features to BERT However, another common method of improving performance is adding linguistic features to the token-level representations. We try a range of linguistic features (described in Section 3.3), that are added in either one or two encoders. We see in the first two columns of results of Table 4 that even though linguistic information sources indeed do improve performance (up to 0.4 absolute), there is no single source that can beat adding just the character-level representations (88.1).

Combining characters and linguistic features An obvious follow-up question is whether we still see improvements for character-level models when also adding linguistic information. In a single encoder, adding characters (third column of results in Table 4) is beneficial for 6 out of 7 linguistic sources (i.e., compared to the first column of results). The scores are, however, not higher than simply adding characters on their own, suggesting that linguistic features are not always beneficial if character-level features are also included. For two encoders, the pattern is less clear, but we do find our highest score thus far when we combine characters and semantic tags (88.4).[12] Using three encoders did not yield clear improvements over two encoders. Therefore, we do not experiment with using more than three encoders.

Robustness to different LMs We want to verify that the character improvements are robust to using different language models (Table 5). We see that adding characters results in improvement for all the LMs under consideration, even for ELMO, which already incorporates characters in creating the initial embeddings. Moreover, combining characters and

	bi-LSTM	Transformer
Char	86.1	79.7
Word	85.3	83.6
GLOVE	85.4	84.6
FASTTEXT	85.5	84.0
ELMO	87.3	84.3
BERT-BASE	**87.6**	**85.4**
BERT-LARGE	87.5	84.7
ROBERTA-BASE	87.0	82.7
ROBERTA-LARGE	86.8	81.9

Table 3: Baseline model for the nine input representations considered, for the bi-LSTM and Transformer architectures. Best score in each column shown in bold.

	No chars		+ characters		
	1-enc	2-enc	1-enc	2-enc	3-enc
BERT	87.6	NA	**88.1**	**88.1**	NA
BERT + word	87.7	87.4	87.8	87.6	86.9
BERT + GLOVE	87.9	87.2	88.1	88.0	86.9
BERT + FASTTEXT	87.8	87.7	87.9	87.9	87.0
BERT + pos	87.6	87.6	87.4	87.6	87.8
BERT + sem	87.9	88.0	88.0	**88.4**	88.1
BERT + lem	87.8	88.0	88.1	88.0	87.4
BERT + dep	87.9	87.5	88.0	87.8	87.8
BERT + ccg	87.8	87.3	87.9	87.8	87.6

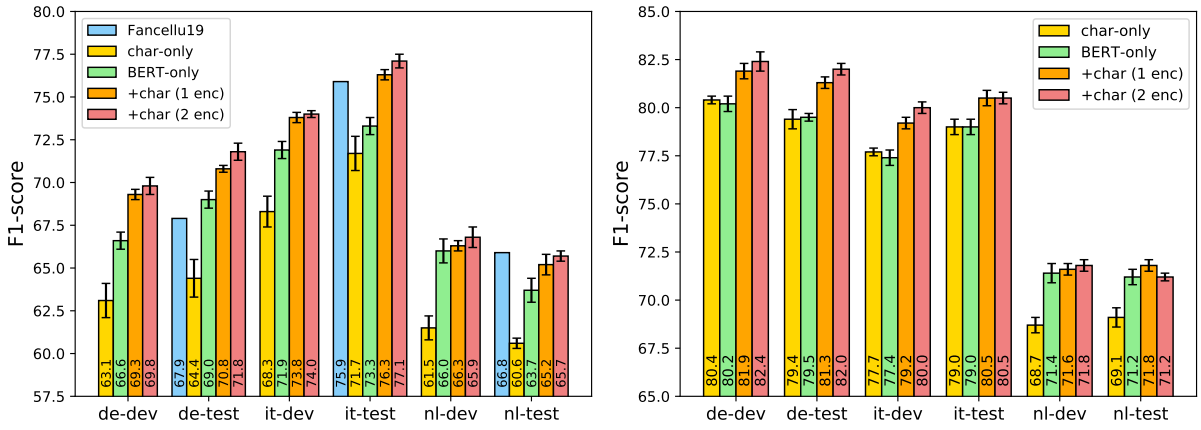
Table 4: Results for adding characters, linguistic information and a combination of the two to the bi-LSTM BERT-BASE model on 3.0.0 English dev.

[9] Standard deviations are omitted for brevity, though available for all experiments here: https://github.com/RikVN/Neural_DRS/

[10] The Transformer models were even tuned longer, since they were more sensitive to small hyperparameter changes.

[11] BERT-BASE significantly outperformed all the other models, except for BERT-LARGE.

[12] This improvement is significant. With gold semantic tags (ceiling performance) we score 88.6.



(a) Scores for PMB release 2.2.0.

(b) Scores for PMB release 3.0.0.

Figure 3: Dev and test scores (F1) for the four models we trained for three languages (German, Italian and Dutch). For 2.2.0, we compare our results to Fancellu et al. (2019).

	No char	+char (1 enc)	+char (2 enc)	+char +sem (2 enc)
ELMO	87.3	87.6	87.8	**88.0**
BERT-BASE	87.6	88.1	88.1	**88.4**
BERT-LARGE	87.5	88.2	87.8	**88.3**
ROBERTA-BASE	87.0	87.3	87.8	**88.0**
ROBERTA-LARGE	86.8	86.8	87.0	**87.3**

Table 5: Results on 3.0.0 English dev of four LMs for adding characters and both characters and semtags.

semantic tags also results in an improvement over just using characters for all the LMs considered.

Robustness across languages We train systems for German, Italian and Dutch for four models: char-only, BERT-ONLY, BERT + char in 1 encoder, and BERT + char in two encoders.[13] The BERT model we use is `bert-multilingual-uncased`. The results for both PMB releases are shown in Figure 3. For all languages, adding characters leads to a clear improvement for both one and two encoders, though for Dutch the improvement is smaller than for German and Italian. Interestingly, the two-encoder setup seems to be preferable for these smaller, non-English data sets. For 2.2.0, we outperform the system of Fancellu et al. (2019) for German and Italian and obtain competitive scores for Dutch.

Comparison to previous work To check whether the improvements hold on unseen data, we run our best models on the test set and compare the scores to previous work (Table 6).[14] We see

[13]We do not train a model that uses semantic tags as features, since there is not enough gold semantic tag data available to train a good tagger for any of these languages.

[14]For the detailed Counter scores see Appendix D.

	2.2.0		3.0.0	
	Dev	Test	Dev	Test
Amateur Boxer	72.2	72.2	78.2	78.8
Pro Boxer	NA	NA	88.2	88.9
Fancellu et al. (2019)	NA	76.4	NA	NA
Evang (2019)	74.4	74.4	NA	NA
van Noord et al. (2018b)	81.2	83.3	84.3	84.9
van Noord et al. (2019)	86.5	86.8	86.8	87.7
Liu et al. (2019b)	85.5	87.1	NA	NA
This work - BERT	85.4	87.9	87.6	88.5
This work - BERT + char (1 enc)	**86.1**	**88.3**	88.1	89.2
This work - BERT + char (2 enc)	85.6	88.1	88.1	89.0
This work - Best model	85.5	87.7	**88.4**	**89.3**

Table 6: Comparison of our four main models to previous work for PMB 2.2.0 and 3.0.0 (English only).

that adding the character-level information has similar (significant) improvements for dev and test on both data sets. The addition of semantic tags might be questionable: for 2.2.0, both the BERT + char models outperform this model, while for 3.0.0 the 0.1 improvement over BERT + char in one encoder is not significant. Despite this, we reach state-of-the-art performance on both data sets, significantly outperforming the previous best scores by van Noord et al. (2019) and Liu et al. (2019b). We also compare to the semantic parser Boxer, which needs input for 6 different PMB layers (Abzianidze et al., 2017). *Amateur Boxer* is trained with internal PMB taggers, while *Pro Boxer* uses the output of a neural multi-task learning system based on BERT (van der Goot et al., 2020). Even though this is an unfair comparison to our system, since the rule-based components of Boxer are (partly) optimized on the dev and test sets, our best model still improves slightly over *Pro Boxer* (significantly on test).

	# Docs	BERT	+char (1 enc)	+char (2 enc)	+ch+sem (2 enc)
All sentences	1,783	88.1	88.7	88.5	**88.8**
Modality	188	86.8	+0.1	+0.1	**+0.4**
Negation	156	88.8	+0.2	-0.1	**+0.4**
Possibility	38	81.3	0.0	+1.0	**+1.5**
Necessity	13	74.5	-1.6	**+1.4**	-0.2
Logical	449	86.3	**+0.7**	+0.2	+0.5
Pronouns	996	88.9	+0.4	+0.4	**+0.6**
Attributes	1,063	87.6	+0.7	+0.4	**+0.8**
Comparatives	45	84.5	**+1.6**	+0.2	-0.2
Named entities	673	88.1	+0.5	+0.3	**+0.6**
Numerals	186	85.8	+1.1	+1.2	**+1.5**

Table 7: F-scores on subsets of sentences that contain a certain phenomenon, based on semantic tags, for the combined dev and test set of PMB release 3.0.0. Full scores shown for BERT and absolute differences for the remaining systems.

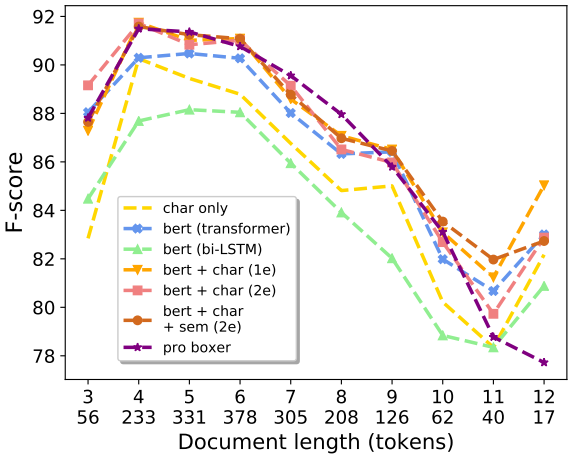


Figure 4: F-scores over document length (tokens) on the combined English dev and test set of 3.0.0. X-axis shows document length (top) and the number of documents for that length (bottom).

5 Analysis

5.1 Semantic tag analysis

We are also interested in finding out *why* the character-level representations help improve performance. As a start, we investigate on what type of sentences and semantic phenomena the character representations are the most beneficial. We introduce a novel method of analysis: selecting subsets of sentences based on the occurrence of certain semantic tags. In the PMB release, each token is also annotated with a semantic tag, which indicates the semantic properties of the token in the given context (Abzianidze and Bos, 2017). This allows us to easily select all sentences that contain certain (semantic) phenomena and evaluate the performance of the different models on those sentences.[15]

The selected phenomena and corresponding F-scores for our four best models (see Table 6) are shown in Table 7.[16] Our best model (+ch+sem) has the best performance on six of the seven phenomena selected, even though the differences are small. The character-level representations seem to help across the board; the +char models improve on the baseline (BERT) in almost all instances.

For *Numerals* and *Named Entities* we expected the characters to help specifically, since (i) BERT representations might not be as optimal for all individual numerals (Wallace et al., 2019), and (ii) the character representations might attend more to capital letters, which often indicate the presence of a named entity. Indeed, the character representations clearly help for *Numerals*, but less so for *Named Entities*. Of course, this analysis only scratched the surface as to why the character-level representations improve performance. We leave a more detailed investigation to future work.

5.2 Sentence length analysis

We are also interested in finding out which model performs well on longer documents. When the Transformer model was introduced, one of the advantages was less decrease in performance for longer sentences (Vaswani et al., 2017). Also, since Boxer is partly rule-based and not trained in an end-to-end fashion, it might be able to handle longer sentences better. Figure 4 shows the performance over sentence length for seven of our trained systems. We see a similar trend for all models: a decrease in performance for longer sentences. We also create a regression model that predicts F-score, with as predictors parser and document length in tokens, similar to van Noord et al. (2018b). We do not find a significant interaction of any model with sentence length, i.e., none of the models decreases significantly less or more than any other model.

To get some idea how well our models would do on longer (possibly multi-sentence) documents, we create a new evaluation set. We select all silver documents with 15 or more and less than 51 tokens that have at least the semtagging or CCG layer marked as gold standard. This resulted in a

[15] Note that this method of analysis can easily be used for other NLP tasks as well, the only requirement being that a semantic tagger has to be used to get the semantic tags.

[16] See Appendix C for the list of semtags per category.

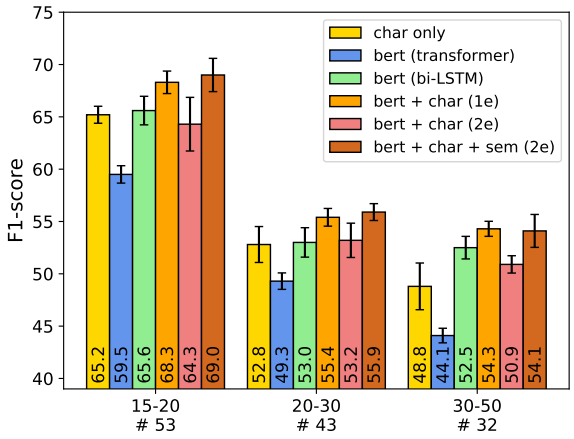


Figure 5: F-scores over document length (tokens) on the silver standard evaluation set of longer documents. X-axis shows the sentence length bins (top) and the number of documents for that length (bottom).

set of 128 DRSs, which should contain the higher quality silver documents. We retrain our models with those sentences removed and plot the performance over sentence length in Figure 5. We see that performance still decreases for longer sentences, though not as much after 30 tokens per document. The Transformer model does not seem to catch up with the bi-LSTM models, even for longer documents. The addition of characters is still beneficial for longer documents, though only in one encoder.

5.3 Discussion

We found that adding character-level representations generally improved performance, though we did not find a clear preference for either the one-encoder or two-encoder model. We believe that, given the better performance of the two-encoder model on the fairly short documents of the non-English languages (see Figure 3), this model is likely the most useful in semantic parsing tasks with single sentences, such as SQL parsing (Zelle and Mooney, 1996; Iyer et al., 2017; Finegan-Dollak et al., 2018), while the one encoder char-CNN model has more potential for tasks with longer sentences/documents, such as AMR (Banarescu et al., 2013), UCCA (Abend and Rappoport, 2013) and GMB-based DRS parsing (Bos et al., 2017; Liu et al., 2018, 2019a). The latter model also has more potential to be applicable for other (semantic parsing) systems as it can be applied to all systems that form token-level representations from a document. In this sense, we hope that our findings here are also applicable for other, more structured, encoder-decoder models developed for semantic parsing (e.g., Yin and Neubig, 2017; Krishnamurthy et al., 2017; Dong and Lapata, 2018; Liu et al., 2019a).

An unexpected finding is that the BERT models outperformed the larger ROBERTA models. In addition, it was even preferable to use BERT only as initial token embedder, instead of fine-tuning using the full model. Perhaps this is an indication that certain NLP tasks cannot be solved by simply training ever larger language models. Moreover, the Transformer model did not improve performance for any of the input representations, while being harder to tune as well. We are a bit hesitant with drawing strong conclusions here, though, since we only experimented with a vanilla Transformer, while recent extensions (e.g., Dehghani et al., 2019; Guo et al., 2019; Press et al., 2020) might be more promising for smaller data sets.

6 Conclusion

We performed a range of experiments on Discourse Representation Structure Parsing using neural sequence-to-sequence models, in which we vary the neural representation of the input documents. We show that, not surprisingly, using pretrained contextual language models is better than simply using characters as input (**RQ1**). However, characters can still be used to improve performance, in both a single encoder and two encoders (**RQ2**). The improvements are larger than using individual sources of linguistic information, and performance still improves in combination with these sources (**RQ3**). The improvements are also robust across different languages models, languages and data sets (**RQ4**) and improve performance across a range of semantic phenomena (**RQ5**). These methods should be applicable to other semantic parsing and perhaps other natural language analysis tasks.

Acknowledgements

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 thank the Center for Information Technology of the University of Groningen for providing access to the Peregrine high performance computing cluster. We also want to thank three anonymous reviewers and our colleagues Lasha Abzianidze, Gosse Minnema, Malvina Nissim, and Chunliu Wang for their comments on earlier versions of this paper.

References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

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*, pages 307–313, Montpellier, France. 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.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Miguel Ballesteros and Yaser Al-Onaizan. 2017. AMR parsing using stack-LSTMs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1269–1275, Copenhagen, Denmark. Association for Computational Linguistics.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal. 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.

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. A platform for collaborative semantic annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 92–96, Avignon, France.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

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. The COLING 2016 Organizing Committee.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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, Palo Alto, CA, USA.

Johan Bos. 2001. Doris 2001: Underspecification, resolution and inference for discourse representation structures. In *ICoS-3 Inference in Compuational Semantics. Workshop Proceedings*, pages 117–124.

Johan Bos. 2008. 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, Venice, Italy.

Johan Bos. 2015. Open-domain semantic parsing with Boxer. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 301–304, Vilnius, Lithuania.

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.

Thorsten Brants. 2000. Tnt: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2019. Core semantic first: A top-down approach for AMR parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3799–3809, Hong Kong, China. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, Online. Association for Computational Linguistics.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017. Doubly-attentive decoder for multi-modal neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1924, Vancouver, Canada. Association for Computational Linguistics.

William B. Cavnar. 1994. Using an n-gram-based document representation with a vector processing retrieval model. In *Proceedings of TREC, NIST special publication*, 500-225, pages 269–277.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany. Association for Computational Linguistics.

Kenneth Ward Church. 1993. Char_align: a program for aligning parallel texts at the character level. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, April 26-30, 2020*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany. Association for Computational Linguistics.

Anna Currey and Kenneth Heafield. 2018. Multi-source syntactic neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2961–2966. Association for Computational Linguistics.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2019. Universal transformers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

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.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–742. Association for Computational Linguistics.

Ted Dunning. 1994. *Statistical identification of language*. Computing Research Laboratory, New Mexico State University Las Cruces, NM, USA.

Long Duong, Hadi Afshar, Dominique Estival, Glen Pink, Philip Cohen, and Mark Johnson. 2017. Multilingual semantic parsing and code-switching. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 379–389, Vancouver, Canada. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Kilian Evang. 2019. Transition-based DRS parsing using stack-LSTMs. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

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*, pages 1422–1426, Seattle, Washington, USA. Association for Computational Linguistics.

Federico Fancellu, Sorcha Gilroy, Adam Lopez, and Mirella Lapata. 2019. Semantic graph parsing with recurrent neural network DAG grammars. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2769–2778, Hong Kong, China. Association for Computational Linguistics.

Christiane Fellbaum, editor. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press, Cambridge, Ma., USA.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Qiankun Fu, Yue Zhang, Jiangming Liu, and Meishan Zhang. 2020. DRTS parsing with structure-aware encoding and decoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6818–6828, Online. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, and Barbara Plank. 2020. Massive choice, ample tasks (MaChAmp):A toolkit for multi-task learning in NLP. *arXiv preprint arXiv:2005.14672*.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Jonas Groschwitz, Matthias Lindemann, Meaghan Fowlie, Mark Johnson, and Alexander Koller. 2018. AMR dependency parsing with a typed semantic algebra. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1831–1841, Melbourne, Australia. Association for Computational Linguistics.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-transformer. 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 1315–1325, Minneapolis, Minnesota. Association for Computational Linguistics.

Fuad Issa, Marco Damonte, Shay B. Cohen, Xiaohui Yan, and Yi Chang. 2018. Abstract Meaning Representation for paraphrase detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 442–452, New Orleans, Louisiana. Association for Computational Linguistics.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 963–973, Vancouver, Canada. Association for Computational Linguistics.

Mark Johnson and Ewan Klein. 1986. Discourse, anaphora and parsing. In *11th International Conference on Computational Linguistics. Proceedings of Coling '86*, pages 669–675, Bonn, Germany.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. An exploration of neural sequence-to-sequence architectures for automatic post-editing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 120–129, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Comput. Linguist.*, 20(3):331–378.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Dan Klein, Joseph Smarr, Huy Nguyen, and Christopher D. Manning. 2003. Named entity recognition with character-level models. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 180–183.

Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. 2017. Neural semantic parsing with type constraints for semi-structured tables. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1526, Copenhagen, Denmark. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields:

Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Phong Le and Willem Zuidema. 2012. Learning compositional semantics for open domain semantic parsing. In *Proceedings of COLING 2012*, pages 1535–1552, Mumbai, India. The COLING 2012 Organizing Committee.

Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. 1990. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404.

Mike Lewis, Kenton Lee, and Luke Zettlemoyer. 2016. LSTM CCG parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 221–231, San Diego, California. Association for Computational Linguistics.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000, Doha, Qatar.

Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–202, Vancouver, Canada. Association for Computational Linguistics.

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)*, pages 429–439, Melbourne, Australia. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2019a. Discourse representation parsing for sentences and documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6248–6262, Florence, Italy. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2019b. Discourse representation structure parsing with recurrent neural networks and the transformer model. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063, Berlin, Germany. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Todor Mihaylov and Anette Frank. 2019. Discourse-aware semantic self-attention for narrative reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2541–2552, Hong Kong, China. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

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

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

Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal*, 7:93–108.

Rik van Noord, Antonio Toral, and Johan Bos. 2019. Linguistic information in neural semantic parsing with multiple encoders. In *Proceedings of the 13th International Conference on Computational Semantics - Short Papers*, pages 24–31, Gothenburg, Sweden. Association for Computational Linguistics.

Eric W. Noreen. 1989. *Computer-intensive Methods for Testing Hypotheses*. Wiley New York.

Fuchun Peng, Dale Schuurmans, Vlado Keselj, and Shaojun Wang. 2003. Language independent authorship attribution with character level n-grams. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary. Association for Computational Linguistics.

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, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany. Association for Computational Linguistics.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *arXiv preprint arXiv:1804.00247*.

Ofir Press, Noah A. Smith, and Omer Levy. 2020. Improving transformer models by reordering their sublayers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2996–3005, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Cicero D Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st international conference on machine learning (ICML-14)*, pages 1818–1826.

Terrence J Sejnowski and Charles R Rosenberg. 1987. Parallel networks that learn to pronounce english text. *Complex systems*, 1(1):145–168.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Gabriel Stanovsky and Ido Dagan. 2018. Semantics as a foreign language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2412–2421, Brussels, Belgium. Association for Computational Linguistics.

Raymond Hendy Susanto and Wei Lu. 2017. Neural architectures for multilingual semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–44, Vancouver, Canada. Association for Computational Linguistics.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024.

Clara Vania, Andreas Grivas, and Adam Lopez. 2018. What do character-level models learn about morphology? The case of dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2573–2583, Brussels, Belgium. 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, Long Beach, California, USA.

Noortje J. Venhuizen, Johan Bos, Petra Hendriks, and Harm Brouwer. 2018. Discourse semantics with information structure. *Journal of Semantics*.

David Vilar, Jan-Thorsten Peter, and Hermann Ney. 2007. Can we translate letters? In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 33–39, Prague, Czech Republic. Association for Computational Linguistics.

Hajime Wada and Nicholas Asher. 1986. BUILDRS: An implementation of DR theory and LFG. In *11th International Conference on Computational Linguistics. Proceedings of Coling '86*, pages 540–545, Bonn, Germany.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? Probing numeracy in embeddings. In *EMNLP-IJCNLP*, pages 5307–5315, Hong Kong, China.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450, Vancouver, Canada. Association for Computational Linguistics.

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, Portland, Oregon, USA.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019a. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94, Florence, Italy. Association for Computational Linguistics.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019b. Broad-coverage semantic parsing as transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3786–3798, Hong Kong, China. Association for Computational Linguistics.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.

A Char-CNN

Figure 6 shows a schematic overview of using the char-CNN (Kim et al., 2016) to encode the word *have* with a width of 2. A width of 2 selects the bigrams *ha*, *av* and *ve*, returning a scalar for each bigram operation, which in turn form a vector $\mathbf{f}_1$ for filter 1. We then take the max value of this vector to obtain the first value of our width 2 (w_2) char-CNN embedding $\mathbf{e}_{\text{w2}_1}$. The final vector $\mathbf{e}_{w2}$ is thus of length n.

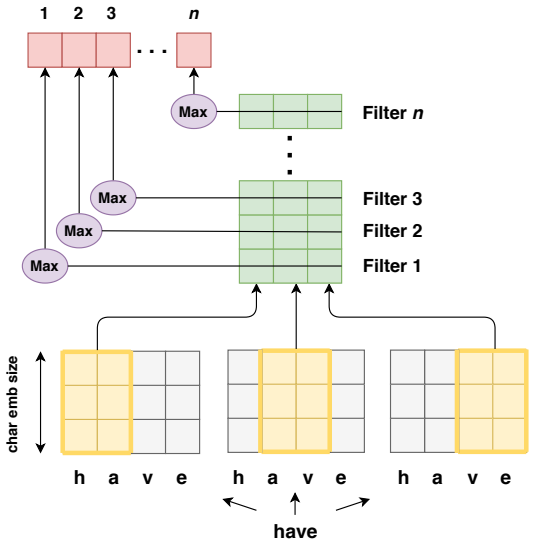


Figure 6: Overview of the char-CNN encoder, encoding the word *have* with bigrams (width = 2) for n filters.

B Experimental settings

Tuning Table 8 gives an overview of the hyperparameters we used and/or experimented with in the tuning stage. This table only gives an overview of the settings for the BERT-BASE model, though the settings for the other representations (described in 3.2) are usually very similar. We performed manual tuning, selecting the settings with the highest F1-score. The number of tuning runs was between 10 and 40 for each representation type and model combination (see Table 3). Output, evaluation (containing F1-scores, standard deviation and confidence interval) and configuration files for our four best models (see Table 6) are available here: https://github.com/RikVN/Neural_DRS/.

Data filtering We filtered ill-formed DRSs from the PMB data sets, which only occurs for silver and bronze data ($< 0.1\%$ of DRSs). For the bi-LSTM models, the filtering of source and target tokens (see Table 8) only filters out three very large documents from training. This was done for efficiency and memory purposes, it did not make a difference in terms of F1-score. However, for the Transformer model this improved F1-score by around 0.5.

Training time and model size A single run of the baseline BERT model takes about 5 hours to train on a single NVIDIA V100 GPU, with around 17 million trainable parameters. Adding character-level representations in one encoder (using the char-CNN) uses around 55 million trainable parameters, with a runtime of around 6 hours. Using a two

encoder setup increases this to around 8 hours, but with only 34 million trainable parameters.

New evaluation set When training models that are evaluated on the silver-standard evaluation set of longer documents, we do not perform fine-tuning on the gold standard data. Also, we run Counter with the `--default-sense` setting (not punishing models that get the word sense wrong), since the word senses of the evaluation set are not gold standard. This has a similar increase of around 1.0 for all models.

Parameter	LSTM	Transf.	Range
Hidden RNN size	300	NA	200 - 600
Decoder RNN size	300	NA	300
Num heads	NA	6	2, 4, 6, 10
hidden_dim	NA	300	300 - 600
ff_hidden_dim	NA	900	300 - 1200
dropout: layer	NA	0.1	0.1, 0.2
residual	NA	0.2	0.1, 0.2
attention	NA	0.1	0.1, 0.2
target emb dim	300	300	300 (GLOVE)
max src tokens	125	50	30 - no max
max tgt tokens	1160	560	300 - no max
layers	1	6	1-3 LSTM, 1-10 Trans
max_norm	3	3	3, 4, 5
scale_grad_by_freq	False	False	True/False
label_smoothing	0.0	0.1	0.0, 0.05, 0.1, 0.2
beam size	10	10	10
max decoding steps	1000	500	500, 1000
scheduled sampling	0.2	0.0	0.0, 0,1, 0.2, 0.3, 0.4
batch size	48	32	12, 24, 32, 48, 64, 128
optimizer	adam	adam	adam, sgd, BertAdam
learining rate	0.001	0.0002	0.0001 - 0.01
grad norm	0.9	0.9	0.7 - 0.95
min target occ	3	3	1, 3, 5, 10, 20

Table 8: An overview of the hyperparameters used for the LSTM and Transformer architecture, that use the BERT-BASE representations. Parameters not specified are left at their default value.

C Semantic tag selection

Modality	NOT NEC POS
Logical	ALT XCL DIS AND IMP BUT
Pronouns	PRO HAS REF EMP
Attributes	QUC QUV COL IST SST
	PRI DEG INT REL SCO
Comparatives	EQU APX MOR LES
	TOP BOT ORD
Named entities	PER GPE GPO GEO ORG ART
	HAP UOM CTC LIT NTH
Numerals	QUC MOY SCO ORD DAT
	DOM YOC DEC CLO

Table 9: Semantic tags that were used to select sentences that contain a certain phenomenon. The example sentence in Table 2 is included in the categories *Modality*, *Pronouns*, *Named Entities* and *Numerals* .

D Detailed scores

Table 10 shows the detailed F-scores for the English dev and test sets of release 2.2.0 and 3.0.0. *Infreq. sense* is the F-score on all concept clauses that were not the most frequent sense for that word in the training set (e.g., `be.v.01`, `like.v.03`). *Perfect sense* is the F-score when we ignore word senses during matching, i.e., `be.v.01` can match with `be.v.02`. The last 9 rows are not in the original detailed Counter scores, but are produced by **DRS-JURY**. Character-level representations help to produce fewer ill-formed and more perfect DRSs, especially on 3.0.0.

E Sentence analysis

Table 11 shows the sentences for which our best model (on 3.0.0 English dev) produced the lowest quality DRSs, with a possible explanation. In Table 12, we show the sentences for which our best model has the best performance (relative to the BERT-ONLY baseline model). It is harder to give an explanation in this case, though we indicate which clauses were (in)correctly predicted by the models.

	PMB release 2.2.0								PMB release 3.0.0							
	Development set				Test set				Development set				Test set			
	bert	+ch (1e)	+ch (2e)	+ch +sem	bert	+ch (1e)	+ch (2e)	+ch +sem	bert	+ch (1e)	+ch (2e)	+ch +sem	bert	+ch (1e)	+ch (2e)	+ch +sem
Prec	87.3	87.8	87.4	87.6	89.8	89.9	89.9	89.5	88.8	88.9	89.3	89.5	90.0	90.6	90.3	90.5
Rec	83.6	84.4	83.6	83.5	86.2	86.7	86.4	86.0	86.4	87.3	86.9	87.2	87.1	87.9	87.6	88.0
F1	85.4	86.1	85.5	85.5	87.9	88.3	88.1	87.7	87.6	88.1	88.1	88.4	88.5	89.2	88.9	89.3
Operators	94.7	95.2	94.7	94.4	94.8	94.7	94.4	94.7	95.0	95.4	95.4	95.7	95.7	95.7	95.7	96.1
Roles	88.0	88.4	88.2	88.0	90.3	90.3	90.5	89.8	89.0	89.0	89.2	89.9	89.4	90.1	89.9	90.0
Concepts	83.9	84.5	84.0	84.8	87.4	87.9	87.6	87.4	84.7	84.9	85.6	85.4	87.3	87.9	87.4	87.7
Nouns	90.8	91.5	91.1	91.4	92.4	92.8	92.4	92.5	90.6	91.0	91.4	91.5	92.0	92.5	91.8	92.5
Verbs	65.6	65.4	64.8	67.6	75.7	76.4	76.3	75.5	69.1	68.9	70.4	69.2	75.3	76.0	76.4	75.3
Adjectives	70.4	74.0	72.7	71.5	70.9	72.3	70.8	71.5	76.1	75.3	76.6	75.5	75.8	77.5	76.2	76.0
Adverbs	90.0	67.7	83.3	63.3	70.0	71.7	73.3	61.0	78.1	77.7	78.7	80.1	88.0	88.2	87.7	88.9
Events	66.7	67.3	66.5	68.4	74.8	75.7	75.4	74.7	70.8	70.5	71.9	70.7	75.4	76.3	76.4	75.4
Perfect sense	87.3	88.1	87.6	87.4	89.3	89.7	89.5	89.1	89.6	90.3	90.2	90.4	91.6	92.2	92.0	92.1
Infreq. sense	50.5	50.5	46.7	52.3	57.2	58.3	58.8	59.1	54.9	57.6	56.5	56.0	62.0	62.8	62.7	63.1
F1 std dev	0.30	0.30	0.17	0.05	0.22	0.22	0.16	0.19	0.19	0.25	0.30	0.34	0.26	0.24	0.29	0.22
F1 confidence	85.0	85.6	85.2	85.4	87.6	88.0	87.9	87.5	87.3	87.8	87.7	87.9	88.2	88.9	88.5	89.0
interval	85.8	86.5	85.7	85.5	88.2	88.6	88.3	88.0	87.9	88.5	88.5	88.8	88.9	89.5	89.4	89.6
# illformed	0.4	0.0	0.2	0.2	0.2	0.0	0.2	0.0	3.2	0.8	2.8	2.0	4.6	3.0	2.8	2.0
# perfect (avg)	235.4	237.4	239.0	239.8	267.0	265.8	266.4	267.2	336.2	350.6	352.4	352.8	358.0	372.4	365.0	367.8
# perfect (all 5)	180	187	183	188	206	213	212	205	212	238	229	226	242	255	239	241
# zero (avg)	4.4	3.4	4.2	4.2	1.6	1.8	1.2	1.8	6.6	3.6	5.0	3.6	5.0	3.2	3.6	2.6
# zero (all 5)	4	3	3	3	1	1	0	1	2	2	1	1	0	0	0	0
# same (all 5)	368	398	379	384	356	368	361	352	347	387	386	365	364	378	361	361

Table 10: Detailed Counter scores for our models on the English dev and test sets of release 2.2.0 and 3.0.0. All scores are averages of 5 runs. Scores are produced by using **DRS-JURY**.

Document	**F1**	**Comment**
Look out!	0.00	Imperative
The dove symbolizes peace.	0.13	Condition + consequence
HBV Union Criticizes Deutsche Bank	0.25	Two multi-word expressions
You can buy stamps at any post office.	0.32	Possibility (can) and quantifier (any)
Fire burns.	0.33	Generic, short
How's Lanzarote?	0.36	How-question
I'd better drive you home.	0.37	Necessity, infrequent sense of drive
What a lot of books! Do they belong to the university library?	0.38	Multi-sentence
Maybe he is Italian or Spanish.	0.40	Possibility and conjunction
I always get up at 6 o'clock in the morning.	0.40	Necessity + clocktime

Table 11: Sentences of the English 3.0.0 dev set for which our best model (+char +sem) produced the **worst** DRSs.

Document	**Diff**	**Comment**
Fish surface for air.	0.554	Correctly produced `Goal`
Oil this bicycle.	0.482	Correctly produced `oil` as a verb
I'm fed up with this winter, I want spring right now!	0.404	Correctly produced `CONTINUATION` and `Pivot`
He's Argentinian.	0.386	BERT-ONLY failed to produce `country` and `Name`
Alas!	0.364	Odd sentence, but correctly produced `state.v.01`
Fire burns.	0.300	Bad performance for both, BERT-ONLY got a score of 0.0
All journeys begin with a first step.	0.300	BERT-ONLY produced a lot of non-matching clauses
How heavy you are!	0.299	BERT-ONLY produced a lot of non-matching clauses
One plus two is equal to three.	0.252	Correctly produced `summation.n.04`
He's not like us.	0.246	Correctly produced `Theme` and `Co-Theme`

Table 12: Sentences of the English 3.0.0 dev set for which our best model (+char +sem) produced the **best** DRSs, relative to the BERT-ONLY baseline.