Linguistic Information in Neural Semantic Parsing with Multiple Encoders

Rik van Noord Antonio Toral Johan Bos
Center for Language and Cognition, University of Groningen
{r.i.k.van.noord, a.toral.ruiz, johan.bos}@rug.nl

Abstract

Recently, sequence-to-sequence models have achieved impressive performance on a number of semantic parsing tasks. However, they often do not exploit available linguistic resources, while these, when employed correctly, are likely to increase performance even further. Research in neural machine translation has shown that employing this information has a lot of potential, especially when using a multi-encoder setup. We employ a range of semantic and syntactic resources to improve performance for the task of Discourse Representation Structure Parsing. We show that (i) linguistic features can be beneficial for neural semantic parsing and (ii) the best method of adding these features is by using multiple encoders.

1 Introduction

Sequence-to-sequence neural networks have shown remarkable performance in semantic parsing (Ling et al., 2016; Jia and Liang, 2016; Konstas et al., 2017; Dong and Lapata, 2018; Liu et al., 2018; Van Noord, Abzianidze, Toral, and Bos, 2018). This architecture is able to learn meaning representations for a range of semantic phenomena, usually without resorting to any linguistic information such as part-of-speech or syntax. Though this is an impressive feat in itself, there is no reason to abandon these resources. Even in machine translation, where models can be trained on relatively large data sets, it has been shown that sequence-to-sequence models can benefit from external syntactic and semantic resources (Sennrich and Haddow, 2016; Aharoni and Goldberg, 2017) and a multi-source approach has proved particularly successful for adding syntax (Currey and Heafield, 2018). The current approaches in neural semantic parsing either include (some) linguistic information in a single encoder (POS-tags in Van Noord and Bos 2017a,b, lemmas in Liu et al. 2018), or use multiple encoders to represent multiple languages rather than linguistic knowledge (Duong et al., 2017; Susanto and Lu, 2017). To our knowledge, we are the first to investigate the potential of exploiting linguistic information in a multi-encoder setup for (neural) semantic parsing.

Specifically, the aims of this paper are to investigate (i) whether exploiting linguistic information can improve semantic parsing and (ii) whether it is better to include this linguistic information in the same encoder or in an additional one. We take as baseline the neural semantic parser for Discourse Representation Structures (DRS, Kamp and Reyle, 1993; Van Noord, Abzianidze, Haagsma, and Bos, 2018) developed by Van Noord, Abzianidze, Toral, and Bos (2018). During encoding we add linguistic information in a multi-encoder setup, including various wide-spread automatic linguistic analyses for the input texts, ranging from lemmatisation, POS-tagging, syntactic analysis, to semantic tagging. We then empirically determine whether using a multi-encoder setup is preferable over merging all input features in a single encoder. The insight gained from these experiments will provide suggestions to improve future neural semantic parsing for DRSs and other semantic formalisms.

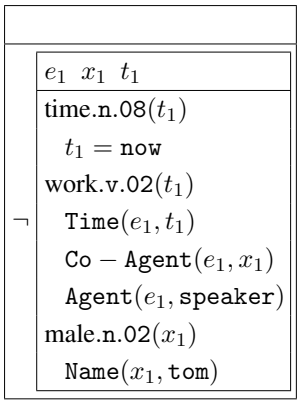
2 Data and Methodology

2.1 Discourse Representation Structures

DRSs are formal meaning representations based on Discourse Representation Theory (Kamp and Reyle, 1993). We use the version of DRT as provided in the Parallel Meaning Bank (PMB, Abzianidze et al. 2017), a semantically annotated parallel corpus, with texts in English, Italian, German and Dutch. DRSs are rich meaning representations containing quantification, negation, reference resolution, comparison operators, discourse relations, concepts based on WordNet, and semantic roles based on VerbNet.

All experiments are performed using the data of the PMB. In our experiments, we only use the English texts and corresponding DRSs. We use PMB release 2.2.0, which contains gold standard (fully manually annotated) data of which we use 4,597 as train, 682 as dev and 650 as test instances. It also contains 67,965 silver (partially manually annotated) and 120,662 bronze (no manual annotations) instances. Most sentences are between 5 and 15 tokens in length. Since we will compare our results mainly to Van Noord, Abzianidze, Toral, and Bos (2018), we will only employ the gold and silver data.

(a)



(b)

```
b1 REF t1
b1 EQU t1 "now"
b1 time "n.08" t1
b2 Time e1 t1
b1 NOT b2
b2 REF e1
b2 Agent e1 "speaker"
b2 work "v.02 e1
b2 Co-Agent e1 x1
b3 REF x1
b3 Name x1 "tom"
b3 male "n.02" x1
```

(c)

```
b2 REF x1
b2 EQU x1 "now"
b2 time "n.08" x1
b1 Time x2 x1
b2 NOT b1
b1 REF x2
b1 Agent x2 "speaker"
b1 work "v.01 x2
b1 Goal x2 x3
b1 REF x3
b1 Name x3 "tom"
```

Figure 1: DRS in box format (a), gold clause representation (b) and example system output (c) for *I am not working for Tom*, with precision of $5/8$ and recall of $5/9$, resulting in an F-score of 58.8.

2.2 Representing Input and Output

We represent the source and target data in the same way as Van Noord, Abzianidze, Toral, and Bos (2018), who represent the source sentence as a sequence of characters, with a special character indicating uppercase characters. The target DRS is also represented as a sequence of characters, with the exception of DRS operators, thematic roles and DRS variables, which are represented as *super characters* (Van Noord and Bos, 2017b), i.e. individual tokens. Since the variable names itself are meaningless, the DRS variables are rewritten to a more general representation, using the De Bruijn index (de Bruijn, 1972). In a post-processing step, the original clause structured is restored.[1]

To include morphological and syntactic information, we apply a lemmatizer, POS-tagger and dependency parser using Stanford CoreNLP (Manning et al., 2014), similar to Sennrich and Haddow (2016) for machine translation. The lemmas and POS-tags are added as a token after each word. For the dependency parse, we add the incoming arc for each word. We also apply the easyCCG parser of Lewis and Steedman (2014), using the supertags.[2] Finally, we exploit semantic information by using semantic tags (Bjerva et al., 2016; Abzianidze and Bos, 2017). Semantic tags are language-neutral semantic categories, which get assigned to a word in a similar fashion as part-of-speech tags. Semantic tags are able to express important semantic distinctions, such as negation, modals and types of quantification. We train a semantic tagger with the TnT tagger (Brants, 2000) on the gold and silver standard data in the PMB release. Examples of the input to the model for each source of information are shown in Table 1.

[1] See Van Noord, Abzianidze, Toral, and Bos (2018) for a more detailed overview of the representation used.

[2] We segment the supertags, e.g. *(S\NP)\(S\NP)* is represented as *(S \ NP) \ (S \ NP)*

Table 1: Example representations for each source of input information.

Source	Representation
Sentence	I am not working for Tom .
Lemma	I be not work for Tom .
POS-tags	PRP VBP RB VBG IN NNP .
Dependency parse	nsubj aux neg ROOT case nmod punct
Semantic tags	PRO NOW NOT EXG REL PER NIL
CCG supertags	NP (S[dcl]\NP)/(S[ng]\NP) (S\NP)\(S\NP) (S[ng]\NP)/PP PP/NP N .

There are two ways to add the linguistic information; (1) merging all the information (i.e., input text and linguistic information) in a single encoder, or (2) using multiple encoders (i.e., encoding separately the input text and the linguistic information). Multi-source encoders were initially introduced for multilingual translation (Zoph and Knight, 2016; Firat et al., 2016; Libovický and Helcl, 2017), but recently were used to introduce syntactic information to the model (Currey and Heafield, 2018). Table 2 shows examples of how the input is structured for using one or more encoders.

Table 2: Example representation when using one or two encoders, for either a single source of information (POS) or multiple sources (POS + Sem) for the sentence *I am not working for Tom.* For readability purposes we show the word-level instead of character-level representation of the source words here.

Source	Encoder	Representation
POS - 1 enc	Enc 1	I PRP am VBP not RB working VBG for IN Tom NNP . .
POS - 2 enc	Enc 1	I am not working for Tom .
	Enc 2	PRP VBP RB VBG IN NNP .
POS + Sem - 1 enc	Enc 1	I PRP PRO am VBP NOW not RB NOT working VBG EXG for IN REL Tom NNP PER . . NIL
POS + Sem - 2 enc	Enc 1	I am not working for Tom .
	Enc 2	PRP PRO VBP NOW RB NOT VBG EXG IN REL NNP PER . NIL

Experiments showed that using more than two encoders drastically decreased performance. Therefore, we merge all the linguistic information in a single encoder (see last row of Table 2).

2.3 Neural Architecture

We employ a recurrent sequence-to-sequence neural network with attention (Bahdanau et al., 2014) and two bi-LSTM layers, similar to the one used by Van Noord, Abzianidze, Toral, and Bos (2018). However, their model was trained with *OpenNMT* (Klein et al., 2017), which does not support multiple encoders. Therefore, we switch to the sequence-to-sequence framework implemented in *Marian* (Junczys-Dowmunt et al., 2018). We use model-type *s2s* (for a single encoder) or *multi-s2s* (for multiple encoders).

For the latter, this means that the multiple inputs are encoded separately by an identical RNN (without sharing parameters). The encoders share a single decoder, in which the resulting context vectors are concatenated. An attention layer[3] is then applied to selectively give more attention to certain parts of the vector (i.e. it can learn that the words themselves are more important than just the POS-tags). A detailed overview of our parameter settings, found after a search on the dev set, can be found in Table 3. When only using gold data, training is stopped after 15 epochs. For gold + silver data, we stop training after 6 epochs, after which we restart the training process from that checkpoint to finetune on only the gold data, also for 6 epochs.

[3]This attention layer is the same for the single source setting.

Table 3: Parameter settings for the Marian seq2seq model, found after a search on the development set. Settings not mentioned are left at default.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
RNN type	LSTM	Dropout RNN	0.2	Learning rate (LR)	0.002	Beam size	10
Enc type	bi-direc	Dropout src/tgt	0.0	LR decay	0.8	Length normalization	0.9
Enc/dec layers	2	Batch size	12	LR decay strategy	epoch	Label smoothing	0.1
Embedding size	300	Optimization crit	ce-mean	LR decay start	9	Skip connections	True
RNN size	300	Vocab size src/tgt	80/150	Clip normalization	3	Layer normalization	True
Epochs	15	Optimizer	adam				

2.4 Evaluation Procedure

Produced DRSs are compared with the gold standard representations by using COUNTER (Van Noord, Abzianidze, Haagsma, and Bos, 2018). This is a tool that calculates micro precision, recall and F-score over matching clauses, similar to the SMATCH (Cai and Knight, 2013) evaluation tool for AMR parsing. All clauses have the same weight in matching, except for REF clauses, which are ignored.

An example of the matching procedure is shown in Figure 1. The produced DRSs go through a strict syntactic and semantic validation process, as described in Van Noord, Abzianidze, Toral, and Bos (2018). If a produced DRS is invalid, it is replaced by a dummy DRS, which gets an F-score of 0.0.

We check whether two systems differ significantly by performing approximate randomization (Noreen, 1989), with $\alpha = 0.05$, $R = 1000$ and $F(model_1) > F(model_2)$ as test statistic for each DRS pair.

3 Results and Discussion

We perform all our experiments twice: (i) only using gold data for training and (ii) with both gold (fully manually annotated) and silver (partially manually annotated) data.

The results of adding external sources of linguistic information are shown in Table 4. We clearly see that using an additional encoder for the linguistic information is superior to merging all the information in a single encoder. For two encoders and only using gold data, the scores increase by at least 0.7 for each source of information individually. Lemmatization shows the highest improvement, most likely because the DRS concepts that need to be produced are often lemmatized versions of the source words. When we stack the linguistic features, we observe an improvement for each addition, resulting in a final 2.7 point F-score increase over the baseline.

If we also employ silver data, we again observe that the multi-encoder setup is preferable over a single encoder, for both isolating and stacking the linguistic features. On isolation, the results are similar to only using gold data, with the exception of the semantic tags, which even hurt the performance now. Interestingly, when stacking the linguistic features, there is no improvement over only using the lemma of the source words.

We now compare our best models to previous parsers[4] (Bos, 2015; Van Noord, Abzianidze, Toral, and Bos, 2018) and two baseline systems, SPAR and SIM-SPAR. As previously indicated, Van Noord, Abzianidze, Toral, and Bos (2018) used a similar sequence-to-sequence model as our current approach, but implemented in OpenNMT and without the linguistic features. Boxer (Bos, 2008, 2015) is a DRS parser that uses a statistical CCG parser for syntactic analysis and a compositional semantics based on λ-calculus, followed by pronoun and presupposition resolution. SPAR is a baseline system that outputs the same DRS for each test instance[5], while SIM-SPAR outputs the DRS of the most similar sentence in the training set, based on a simple word embedding metric.[6] The results are shown in Table 5. Our model clearly outperforms the previous systems, even when only using gold standard data. When compared to Van Noord, Abzianidze, Toral, and Bos (2018), retrained with the same data used in our systems,

[4]Since Liu et al. (2018) used data from the Groningen Meaning Bank instead of the PMB, we cannot make a comparison.

[5]For PMB release 2.2.0 this is the DRS for *Tom voted for himself.*

[6]See Section 5.1 of Van Noord, Abzianidze, Haagsma, and Bos (2018) for an explanation of the high baseline scores.

Table 4: Table (a) and (b) show the results of adding **a single type** of linguistic information. Table (c) and (d) show the results for **stacking multiple types** of linguistic information. Reported scores are F-scores on the development set, averaged over 5 runs of the system, with confidence scores.

(a)	**Gold only: single type**	
Model	**1 enc**	**2 enc**
Baseline	78.6 ± 0.6	NA
POS-tags	**79.5** ± 0.8	79.3 ± 0.6
Semantic tags	79.0 ± 0.9	79.3 ± 0.4
Lemma	78.6 ± 0.4	**79.9** ± 0.4
Dependency parse	78.9 ± 0.7	79.3 ± 0.8
CCG supertags	78.6 ± 1.1	79.4 ± 0.9

(b)	**Gold + silver: single type**	
Model	**1 enc**	**2 enc**
Baseline	84.5 ± 0.3	NA
POS tags	**84.8** ± 0.3	84.9 ± 0.4
Semantic tags	83.5 ± 0.6	84.0 ± 0.4
Lemma	84.0 ± 0.2	**85.6** ± 0.4
Dependency parse	83.9 ± 0.4	84.6 ± 0.3
CCG supertags	83.8 ± 0.3	84.8 ± 0.5

(c)	**Gold only: stacking**	
Model	**1 enc**	**2 enc**
Baseline	78.6 ± 0.6	NA
+ Lemma	78.6 ± 0.4	79.9 ± 0.4
+ Semantic tags	**79.4** ± 0.6	80.5 ± 0.6
+ POS tags	79.4 ± 0.3	80.8 ± 0.3
+ CCG supertags	79.4 ± 0.6	81.0 ± 0.6
+ Dependency parse	78.8 ± 0.7	**81.3** ± 0.9

(d)	**Gold + silver: stacking**	
Model	**1 enc**	**2 enc**
Baseline	**84.5** ± 0.3	NA
+ Lemma	84.0 ± 0.2	**85.6** ± 0.4
+ POS-tags	84.3 ± 0.4	85.5 ± 0.3
+ CCG supertags	84.5 ± 0.2	85.6 ± 0.6
+ Dependency parse	84.5 ± 0.2	85.4 ± 0.4
+ Semantic tags	83.7 ± 0.4	85.1 ± 0.2

the largest improvement (3.6 and 3.5 for dev and test) comes from switching framework and changing certain parameters such as the optimizer and learning rate. However, the linguistic features are clearly still beneficial when using only gold data (increase of 2.7 and 1.9 for dev and test), and also still help when employing additional silver data (1.1 and 0.3 increase for dev and test, both significant).

Table 5: Results on the test set compared to a number of baseline parsers and the Seq2seq OpenNMT model of Van Noord, Abzianidze, Toral, and Bos (2018). Our scores are averages of 5 runs, with confidence scores.

	Dev			**Test**		
	Prec	**Rec**	**F-score**	**Prec**	**Rec**	**F-score**
SPAR	42.3	37.9	40.0	44.4	37.8	40.8
SIM-SPAR	52.4	54.2	53.3	57.0	58.4	57.7
Boxer (Bos, 2015)	72.5	72.0	72.2	72.1	72.3	72.2
Van Noord, Abzianidze, Toral, and Bos (2018)	83.5	78.5	80.9	85.0	81.4	83.2
This paper: gold only	81.9	75.6	78.6 ± 0.6	85.1	78.1	81.5 ± 0.2
This paper: gold only + all ling	84.3	78.5	81.3 ± 0.9	86.6	80.4	83.4 ± 0.4
This paper: gold + silver	85.9	83.2	84.5 ± 0.3	87.4	86.0	86.7 ± 0.2
This paper: gold + silver + lemma	86.5	84.8	85.6 ± 0.4	87.6	86.3	87.0 ± 0.4

4 Conclusions

In this paper we have shown that a range of linguistic features can improve performance of sequence-to-sequence models for the task of parsing Discourse Representation Structures. We have shown empirically that the best method of adding these features is by using a multi-encoder setup, as opposed to merging the sources of linguistic information in a single encoder. We believe that this method can also be beneficial for other semantic parsing tasks in which sequence-to-sequence models do well.

References

Abzianidze, L., J. Bjerva, K. Evang, H. Haagsma, R. van Noord, P. Ludmann, D.-D. Nguyen, and J. Bos (2017, April). 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*, Valencia, Spain, pp. 242–247. Association for Computational Linguistics.

Abzianidze, L. and J. Bos (2017, September). Towards universal semantic tagging. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS 2017) – Short Papers*, Montpellier, France, pp. 307–313. Association for Computational Linguistics.

Aharoni, R. and Y. Goldberg (2017). Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 132–140. Association for Computational Linguistics.

Bahdanau, D., K. Cho, and Y. Bengio (2014). Neural machine translation by jointly learning to align and translate. *CoRR abs/1409.0473*.

Bjerva, J., B. Plank, and J. Bos (2016). Semantic tagging with deep residual networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan, pp. 3531–3541.

Bos, J. (2008). Wide-coverage semantic analysis with Boxer. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, Volume 1 of *Research in Computational Semantics*, pp. 277–286. Venice, Italy: College Publications.

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

Brants, T. (2000). Tnt: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, Stroudsburg, PA, USA, pp. 224–231. Association for Computational Linguistics.

Cai, S. and K. Knight (2013, August). 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)*, Sofia, Bulgaria, pp. 748–752. Association for Computational Linguistics.

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

de Bruijn, N. G. (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, pp. 381–392. Elsevier.

Dong, L. and M. 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)*, pp. 731–742. Association for Computational Linguistics.

Duong, L., H. Afshar, D. Estival, G. Pink, P. Cohen, and M. Johnson (2017). Multilingual semantic parsing and code-switching. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 379–389.

Firat, O., K. Cho, and Y. Bengio (2016). Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of NAACL-HLT*, pp. 866–875.

Jia, R. and P. 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, Berlin, Germany, pp. 12–22.

Junczys-Dowmunt, M., R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins, and A. Birch (2018, July). Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, Melbourne, Australia, pp. 116–121. Association for Computational Linguistics.

Kamp, H. and U. Reyle (1993). *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Dordrecht: Kluwer.

Klein, G., Y. Kim, Y. Deng, J. Senellart, and A. Rush (2017). OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, Vancouver, Canada, pp. 67–72. Association for Computational Linguistics.

Konstas, I., S. Iyer, M. Yatskar, Y. Choi, and L. Zettlemoyer (2017, July). 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)*, Vancouver, Canada, pp. 146–157. Association for Computational Linguistics.

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

Libovický, J. and J. 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)*, pp. 196–202. Association for Computational Linguistics.

Ling, W., P. Blunsom, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, F. Wang, and A. 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, Berlin, Germany, pp. 599–609.

Liu, J., S. B. Cohen, and M. 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, Melbourne, Australia, pp. 429–439.

Manning, C., M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60.

Noreen, E. W. (1989). *Computer-intensive Methods for Testing Hypotheses*. Wiley New York.

Sennrich, R. and B. Haddow (2016). Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, Volume 1, pp. 83–91.

Susanto, R. H. and W. 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)*, Volume 2, pp. 38–44.

Van Noord, R., L. Abzianidze, H. Haagsma, and J. Bos (2018). Evaluating scoped meaning representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, pp. 1685–1693. European Language Resources Association (ELRA).

Van Noord, R., L. Abzianidze, A. Toral, and J. Bos (2018). Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics 6*, 619–633.

Van Noord, R. and J. Bos (2017a). Dealing with co-reference in neural semantic parsing. In *Proceedings of the 2nd Workshop on Semantic Deep Learning (SemDeep-2)*, Montpellier, France, pp. 41–49.

Van Noord, R. and J. Bos (2017b). Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal 7*, 93–108.

Zoph, B. and K. Knight (2016). Multi-source neural translation. In *Proceedings of NAACL-HLT*, pp. 30–34.