Neural Boxer at the IWCS Shared Task on DRS Parsing

Rik van Noord
Center for Language and Cognition, University of Groningen
r.i.k.van.noord@rug.nl

Abstract

This paper describes our participation in the shared task of Discourse Representation Structure parsing. It follows the work of Van Noord et al. (2019), who employed a neural sequence-to-sequence model to produce DRSs, also exploiting linguistic information with multiple encoders. We provide a detailed look in the performance of this model and show that (i) the benefit of the linguistic features is evident across a number of experiments which vary the amount of training data and (ii) the model can be improved by applying a number of postprocessing methods to fix ill-formed output. Our model ended up in second place in the competition, with an F-score of 84.5.

1 Introduction

Semantic parsing is the problem of mapping natural language utterances to interpretable meaning representations. Specifically, we focus on producing Discourse Representation Structures (DRS), based on Discourse Representation Theory (Kamp, 1984; Kamp and Reyle, 1993), a formalism developed in formal semantics. Since DRS parsing is a complex task, among others dealing with scope, negation, presuppositions and discourse structures, previous parsers used to be a combination of symbolic and statistical components (Bos, 2008, 2015). However, recently neural sequence-to-sequence models achieved impressive performance on the task (Liu et al., 2018; Van Noord et al., 2018, 2019).

This work describes our system with which we participated in the first shared task on DRS parsing (Abzianidze et al., 2019). Our system is the same as described in Van Noord et al. (2019), except for a number of additional post-processing methods to solve ill-formed DRSs. Van Noord et al. (2019) follow Van Noord et al. (2018) in employing a sequence-to-sequence neural network to produce the DRSs, using character-level input[1] and rewriting the variables to a more general structure. They then improve on this work by exploiting linguistic information (POS-tags, semantic tags[2], dependency parses, CCG categories and lemmas), using a second encoder to provide this information to the model. We first demonstrate how sensitive the model is to changes in certain parameter settings, after which we determine if the model could still benefit from additional gold or silver standard data (Section 2). In Section 3, we describe our new postprocessing methods to decrease the number of ill-formed DRSs. Finally in Section 4 we perform a detailed error analysis.

2 Analysis

All results are obtained by training on the data released in Parallel Meaning Bank release 2.2.0 (Abzianidze et al., 2017). This release contains gold standard (fully manually annotated) data of which we use 4,597 as train, 682 as dev and 650 as test instances. Moreover, we also use the 67,965 silver (partly manually annotated) instances as extra training data for some experiments. Reported values are F-scores calculated by COUNTER (Van Noord, Abzianidze, Haagsma, and Bos, 2018), which are averaged over 3

[1] They use *super characters* (Van Noord and Bos, 2017a,b) for DRS roles and operators.

[2] See Bjerva et al. (2016) and Abzianidze and Bos (2017) for a detailed description of semantic tagging.

different runs of the system. In this section, all experiments are performed on the development set. All code is publicly available.[3]

2.1 Parameter sensitivity

The parameter search in Van Noord et al. (2019) was performed by applying a hill climbing method: one parameter was tuned with all other parameters fixed. Only if a parameter returned a significant improvement, it was chosen over the current setting. In Van Noord et al. (2019) we only did a single pass over all parameters, meaning that there is likely still room for improvement. Also, it is interesting to determine the sensitivity of our model to certain parameter settings, since we generally prefer a model that is not too sensitive to slight changes in them.

We performed a detailed search for the hyper-parameters RNN dimension, dropout, learning rate and beam size, of which the results are shown in Figure 1.[4] We see that the model allows for some variety in dropout and learning rate, though there is a sharp decrease in performance if we move too far from the optimal setting. For the RNN dimension there is not much difference if more than 250 nodes are used, though increasing the dimension to 500 could improve performance. Similarly, a beam size of 5 is sufficient for good performance, but using a beam size of 15 could be an extra small gain.

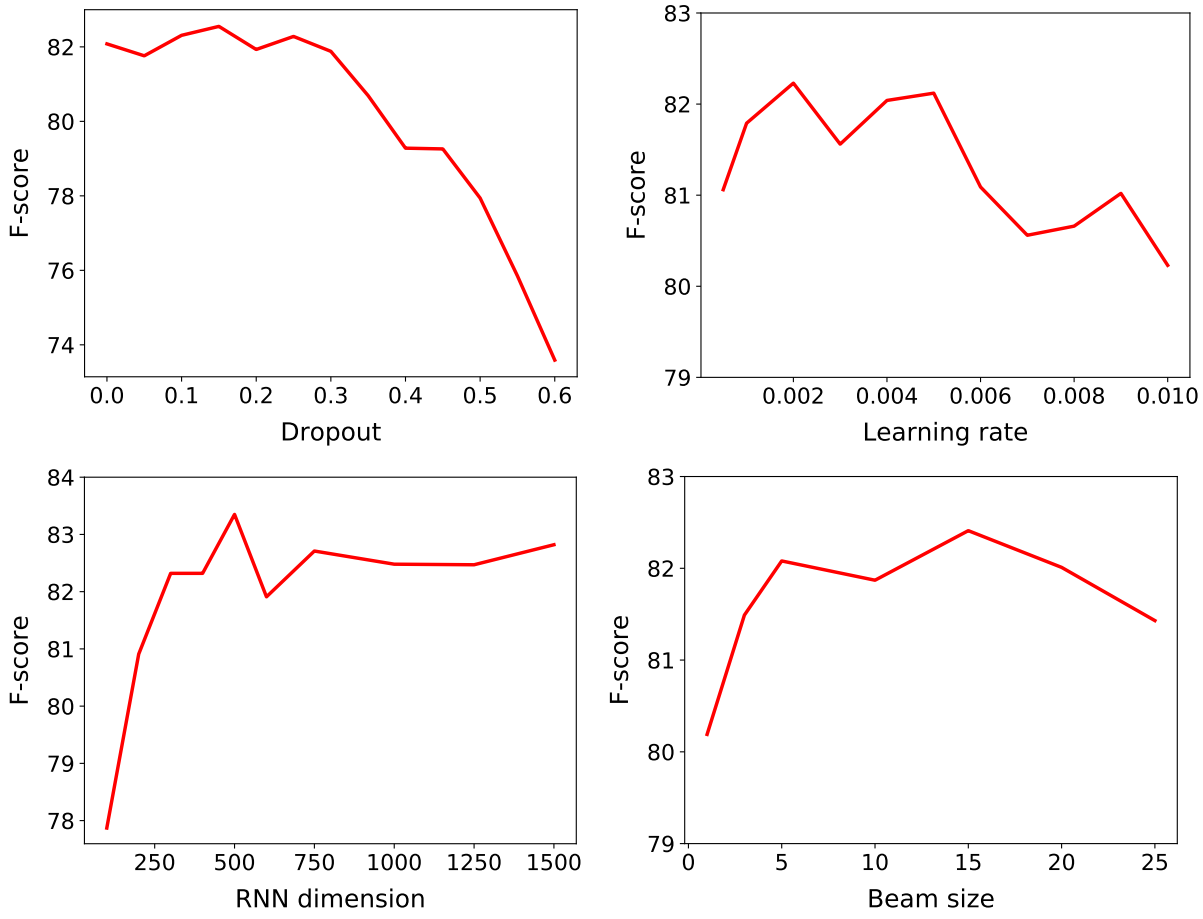

Figure 1: Performance of our best model trained on only gold standard data for the hyper parameters (shared task setting in brackets): RNN dimensions (300), dropout (0.2), learning rate (0.002) and beam size (10).

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

[4] Performed after the shared task deadline, our official scores are with the parameters reported in Van Noord et al. (2019).

2.2 Learning curves

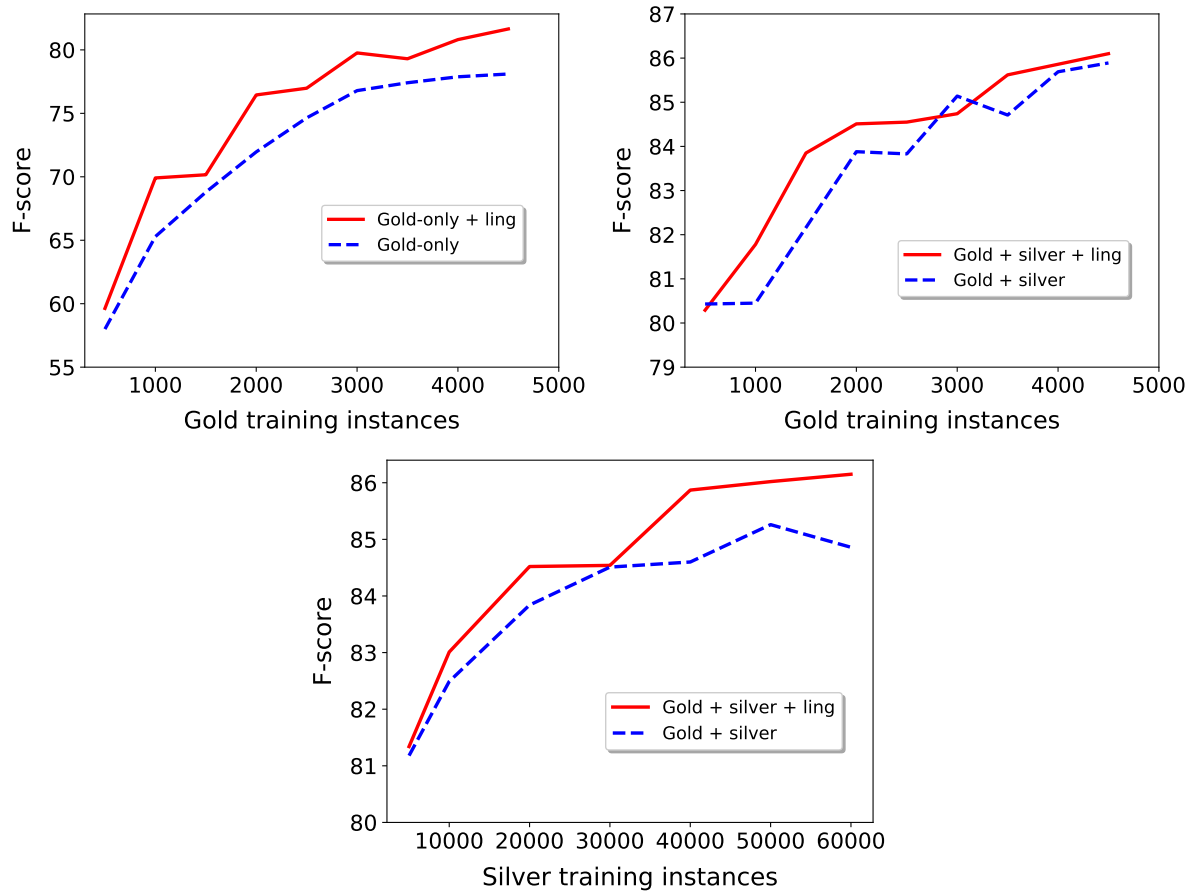

Figure 2: Performance of our baseline and best system for: training on only different amounts of gold data (top left), training on all silver standard data with different amounts of gold data (top right), and training on different amounts of silver data, finetuned on all gold data (bottom).

Since manual DRS annotation is a hard and time consuming task, it is interesting to know how much we can still benefit from extra silver and gold standard examples, as well as identifying how the amount of data contributes to the final scores. Concurrently, we can also observe the impact of the linguistic features across different amounts of training data. We show the learning curves of two models: the baseline model and the best model. The best model employs the linguistic features in a separate encoder.

How much gold is needed?
The top-left graph of Figure 2 (left) shows the performance of our two models when only using gold training data. It is clear that additional gold data still improves performance, though in a way we already knew this due to the success of employing silver data. Therefore, we also plot the effect of varying the amount of gold data used when using all available silver data in the top right graph of Figure 2. We see the same trend there: adding gold standard data still clearly improves performance.

How much silver is needed?
The effect of varying the amount of silver data is shown in bottom graph of Figure 2. The initial addition of silver data is clearly beneficial. However, the effect seems to diminish a bit for the best model after 40,000 silver instances. The baseline model, though, still improves after 40,000 instances. In general, it seems like additional silver data could be beneficial, though the extra benefit is likely to be small.

Impact of linguistic features
For all experiments shown in Figure 2, we see that the linguistic features increase performance across virtually all amounts of data used. This further confirms that the linguistic features supply the model with additional useful information.

3 Postprocessing methods

There are a number of syntactic and semantic requirements for a set of clauses to be considered a well-formed DRS (Van Noord et al., 2018). Among others, there should be a single main box and there should be no loop in the subordination of boxes. Since we do not restrict our model when producing clauses, these errors can occur. However, they can often be (easily) fixed, by changing a single clause or a set of clauses. We outline a few of those methods below. Moreover, we propose a method to improve performance on word sense disambiguation.

3.1 Improvement methods

Removing clauses
One of the problems of the neural model is that it can get stuck in a loop producing (more or less) the same output. We apply two straightforward methods to fix these instances. For one, we remove all clauses after clause number 75. Second, we remove clauses of concepts, roles and operators (except `REF`) if they occur more than three times. We only look at the second argument of a clause (the identifier), for example, if a full DRS contains five `Theme` clauses, we remove the last two, no matter the other values in those clauses.

No main box found
If there is no main box found, this means that there are multiple independent boxes. For two independent boxes, we first try to remedy this by changing a single discourse variable in one of these boxes. We change a discourse variable that is unique in **b1** to a unique variable in **b2** (and vice versa), to establish a connection between the boxes. For each of these possible changes we check whether the DRS is well-formed now, and if so, return the new DRS. For multiple independent boxes, and if the previous method failed for two independent boxes, we start merging boxes together (e.g. replace each occurrence of **b1** by **b2**), until we find a well-formed DRS. If this does not result in a well-formed DRS, we return a non-matching dummy DRS.[5]

Subordinate relation has a loop
This can occur if boxes indirectly subordinate themselves, e.g. **b0** subordinates **b1**, **b1** subordinates **b2** and **b2** subordinates **b0**. To solve this, we first try to merge the offending box with each of the other boxes in the DRS. If this does not work, we try to remove the offending box from the DRS. If the DRS is still ill-formed, we start the process again if the error is *Subordinate relation has a loop* (but now with the offending box removed) or apply the previously described fix for *No main box found*. A non-matching dummy DRS is returned if the DRS is still ill-formed after these steps.[5]

Fixing senses
Previous work showed that the neural model often produced the wrong word sense for the correct concept (Van Noord et al., 2018). It even often output senses that were never observed in the training set. We apply a simple method to fix these instances. If a concept + sense is not present in the gold standard training set, we replace the sense by the most frequent sense for this concept in the training set. For example, we change `grow.v.01` to `grow.v.07` and `fast.n.02` to `fast.a.02`. Note that this method does not influence whether a DRS is well-formed or not. This was implemented after the shared task deadline, meaning our final shared task system did not apply this method.

[5]The shared task system returned the SPAR default DRS.

3.2 Results

We can check by how much the scores in Van Noord et al. (2019) would have improved if these methods had been applied. The results of adding the improvement methods incrementally are shown in Table 1. We see that simply removing clauses returns only modest gains, but fixing ill-formed DRSs gives a substantial improvement, even for our best model. By applying these fixes on the shared task evaluation set we decreased the number of ill-formed DRSs from 283 to 9, but since this evaluation set was not released, we do not know the impact on the F-score. The method for fixing word senses also proved quite effective, improving the final F-score by 0.2 to 0.4.

	Initial		**+ Removing clauses**		**+ Solving ill-formed**		**+ Fixing senses**	
	ill (%)	F1	ill (%)	F1	ill (%)	F1	ill (%)	F1
Gold-only baseline	2.6	78.6	2.6	78.8	0.2	79.5	0.2	79.8
Gold-only + ling	2.7	81.3	2.7	81.4	0.1	82.2	0.1	82.4
Gold + silver + ling	1.5	84.5	1.5	84.5	0.0	85.1	0.0	85.4
Gold + silver + ling	0.9	85.6	0.9	85.7	0.0	86.1	0.0	86.4

Table 1: Impact of the new postprocessing methods on the dev set results of Van Noord et al. (2019).

4 Results & Error Analysis

4.1 Detailed F-scores

The right column of Table 2 shows the results of our official submission to the shared task. We obtained an F-score of 84.5, with a precision and recall of 85.5 and 83.6, which is slightly lower than our dev and test scores. We ended up in second place in the competition, though the difference with the first place (84.8) was not statistically significant.

Table 2 also contains the automatically calculated detailed F-scores on the test set for both Van Noord et al. (2018) and Van Noord et al. (2019). The improvement for the new neural model and the addition of the linguistic features mainly comes from improved performance on the roles and concepts. It is evident that word sense disambiguation is a hard problem, since even with our method to fix word senses, we still obtain large increases in F-score for oracle senses. The improvement methods described in Section 3 result in a modest, but significant 0.4 increase on the test set.

	NeuDRS-18	**NeuDRS-19**	**This work - test**	**This work - eval**
All clauses	83.2	87.0	87.4	84.5
DRS Operators	93.4	94.4	94.5	94.2
VerbNet roles	83.2	87.1	87.3	83.5
WordNet synsets	79.7	84.3	85.2	82.3
nouns	85.5	89.3	90.1	87.5
verbs	65.6	72.2	73.0	68.9
adjectives	64.7	70.4	73.6	74.2
adverbs	50.0	72.6	71.9	45.5
Oracle sense numbers	85.7	89.2	89.6	87.1
Oracle synsets	90.0	92.6	92.7	90.7
Oracle roles	86.7	90.0	90.4	88.5

Table 2: F-scores of fine-grained evaluation on the test set of the work of Van Noord et al. (2018) (NeuDRS-18) , Van Noord et al. (2019) (NeuDRS-19) and this work, which is NeuDRS-19 plus the improvement methods described in Section 3.

4.2 Examples

Relatively good performance	Relatively bad performance
(a) Tom died when he was 97.	(f) These bananas are not ripe.
(b) I read comic books.	(g) A book about dancing is lying on the desk.
(c) We should drink 64 ounces of fluids a day.	(h) Approximately seven billion people inhabit our planet.
(d) I look down on liars and cheats.	(i) The trip will take approximately five hours.
(e) You can't run away.	(j) Tom was too tired to speak.

Table 3: Sentences for which our model performed relatively well and poor.

The organizers provided us with five sentences in the test set for which our model did relatively well, and 5 for which the model performed relatively poor. The sentences are shown in Table 3.[6] Interestingly, the model does well on sentences containing numbers, (a) and (c), but fails to capture the correct interpretation of *approximately* in (h) and (j). For (b) and (d), our model correctly identified the multi-word expressions *comic_book* and *look_down_on*, perhaps due to the character-level input. For (e), our model produced a perfect DRS, while other approaches had trouble either producing *run_away*, or the fact that the sentence was addressed to a hearer. Sentence (j) is interpreted in the gold standard as *Tom could not speak because he was tired*, which our model simply failed to capture. It produced a DRS more similar to the meaning of *Tom spoke and was tired*, as is shown in Table 4.

Gold standard		Produced output	Matching
b2 REF x1	b6 DRS b1	b1 REF x1	b2 ⇒ b1
b2 Name x1 "tom"	b6 DRS b3	b1 Name x1 "tom"	x1 ⇒ x1
b2 male "n.02" x1	b6 RESULT b1 b3	b1 male "n.02" x1	b1 ⇒ b2
b1 REF t1	b1 REF s1	b2 REF x2	t1 ⇒ x2
b1 TPR t1 "now"	b1 Theme s1 x1	b2 TPR x2 "now"	s1 ⇒ x3
b1 Time s1 t1	b1 tired "a.01" s1	b2 Time x3 x2	b3 ⇒ ∅
b1 time "n.08" t1	b5 REF e1	b2 time "n.08" x2	b4 ⇒ ∅
b3 NOT b4	b5 Agent e1 x1	b2 REF x3	b5 ⇒ ∅
b4 POS b5	b5 speak "v.01" e1	b2 Theme x3 x1	b6 ⇒ ∅
		b2 tired "a.01" x3	e1 ⇒ ∅
		b2 Co-Theme x3 x4	∅ ⇒ x4
		b2 REF x4	
		b2 speak "n.01" x4	

Table 4: Gold standard and produced DRS for the sentence *Tom was too tired to speak.* This gave us a precision and recall of $7/9$ and $7/14$, resulting in an F-score of 0.61.

5 Conclusion

This paper provided a more detailed analysis on the study of Van Noord et al. (2019), in which they showed that explicitly encoder linguistic features can be beneficial for neural sequence-to-sequence models on the task of DRS parsing. We show that the benefit of these features are robust across experiments with different amounts of training data. Moreover, we show that the model is not too sensitive to small variations in parameter settings, perhaps even observing room for more finetuning of the model. Lastly, we show that a number of rule-based methods can drastically decrease the number of ill-formed DRSs. Our method ultimately obtained a second place in the shared task competition with an F-score of 84.5.

[6] DRSs available at: https://urd2.let.rug.nl/ rikvannoord/DRS/IWCS/

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.

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.

Abzianidze, L., R. van Noord, H. Haagsma, and J. Bos (2019). The first shared task on discourse representation structure parsing. In *Proceedings of the IWCS 2019 Shared Task on Semantic Parsing*.

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.

Kamp, H. (1984). A theory of truth and semantic representation. In J. Groenendijk, T. M. Janssen, and M. Stokhof (Eds.), *Truth, Interpretation and Information*, pp. 1–41. Dordrecht – Holland/Cinnaminson – U.S.A.: FORIS.

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

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.

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.

Van Noord, R., A. Toral, and J. Bos (2019). Linguistic information in neural semantic parsing with multiple encoders. In *IWCS 2019-13th International Conference on Computational Semantics-Short papers*.