Tüpa at SemEval-2019 Task 1: (Almost) feature-free Semantic Parsing

Tobias Pütz
Department of Linguistics
University of Tübingen
SFB 833 A3

Kevin Glocker
Department of Linguistics
University of Tübingen

{tobias.puetz,kevin.glocker}@student.uni-tuebingen.de

Abstract

Our submission for Task 1 'Cross-lingual Semantic Parsing with UCCA' at SemEval-2018 is a feed-forward neural network that builds upon an existing state-of-the-art transition-based directed acyclic graph parser. We replace most of its features by deep contextualized word embeddings and introduce an approximation to represent non-terminal nodes in the graph as an aggregation of their terminal children. We further demonstrate how augmenting data using the baseline systems provides a consistent advantage in all open submission tracks. We submitted results to all open tracks (English, in- and out-of-domain, German in-domain and French in-domain, low-resource). Our system achieves competitive performance in all settings besides the French, where we did not augment the data. Post-evaluation experiments showed that data augmentation is especially crucial in this setting.

1 Introduction

Semantic Parsing is the task of assigning an utterance a structured representation of its meaning. The goal is to assign similar structures to utterances with similar meanings, regardless of their syntactic realizations. In Syntactic Parsing, for instance, the sentence 'John saw Paul.' will have a different structure than 'Paul was seen by John'. Semantic Parsing, in contrast, aims to solely encode the fact that John saw Paul. Deriving a semantic representation of an utterance has various applications. It can serve as a starting point for the evaluation of machine translation systems, as the structure of the semantic representation should be similar across languages. Birch et al. (2016) use human annotated scores of individual UCCA semantic units in their HUME metric to provide a fine-grained analysis of translation quality and improve scalability to longer sentences by approximating human judgement semi-automatically from the annotated scores of each unit. Explicit semantic representations could also provide the structured information necessary to alleviate recent issues in Natural Language Inference (NLI) where McCoy and Linzen (2019) showed that state-of-the-art NLI systems fail to recognize that e.g. 'Alice believes Mary is lying.' does not entail 'Alice believes Mary.'. Using precise semantic representations of the sentences a theorem could be built on which various logical inferences can be performed with a theorem prover such as in Martínez-Gómez et al. (2016).

Universal Conceptual Cognitive Annotation (UCCA) (Abend and Rappoport, 2013) is a semantic grammar formalism where natural language expressions are analyzed as deep directed acyclic graph (DAG) structures, deep meaning the graphs feature non-terminal nodes. Due to it's coarse-grained representation using cognitively motivated categories it is both domain and language independent and quickly learned even by annotators without a linguistic background (Abend and Rappoport, 2013).

The goal of the SemEval-2018 Task 1 'Cross-lingual Semantic Parsing with UCCA' was to develop a parser producing UCCA-DAG structures trained on articles from Wikipedia in English and passages from the book "Twenty Thousand Leagues Under the Sea" in French and German. The parsers were evaluated on the DAG-F1 metric on in-domain passages in English, French and German as well as out-of-domain passages in English in both an open and a closed track (Hershcovich et al., 2018b). Since we made extensive use of external resources we participated only in the open track of all settings.

For our participation, we build upon the transition-based DAG parser Tupa (Hershcovich et al., 2017). Our adaptation reuses the transition

system and oracle. We extend Tupa with respect to its representations of non-terminal nodes in a way that they are an aggregation of all their terminal nodes. While Tupa uses a Recurrent Neural Network, our system is a simple feed-forward network that uses a small set of features and ELMo contextualized embeddings (Peters et al., 2018) made available by Che et al. (2018)[1] and Fares et al. (2017).

2 Background

Until recently, semantic parsers were exclusively symbolic rule-based systems (Bos, 2005). These systems rely on complex hand-written and necessarily language-specific sets of rules, requiring a re-implementation for every new language. More recently, neural methods have also arrived in the domain of Semantic Parsing. They achieve state-of-the-art results while being largely language-agnostic. Since these systems usually require large amounts of annotated data, this line of work is largely concerned with the augmentation of training data. Hershcovich et al. (2018a) recognize the similarity between several annotation schemes and jointly learn to parse other semantic formalisms in a multi-task setting, while van Noord et al. (2018) add large amounts of automatically annotated data to their training data. Both approaches led to significant improvements over not using the additional data.

3 Silver Data

We created additional training data for both English and German using the open track baseline systems. The English silver data was taken from the 1B word benchmark (Chelba et al., 2014), the German from the archive of the newspaper taz. For both languages, we took the first 15,000 sentences of the corpora and added UCCA annotation using the baseline systems. Our training datasets then consisted of the concatenation of gold and silver data, and another gold only set. Due to a lack of time we did not create silver data for our French submission. Post-evaluation results for French, trained on v2.0 of the GSD treebank[2] provided by Universal Dependencies (Nivre et al., 2016), are presented in Section 6.1.

[1] https://github.com/HIT-SCIR/ELMoForManyLangs/

[2] https://universaldependencies.org/treebanks/fr_gsd/

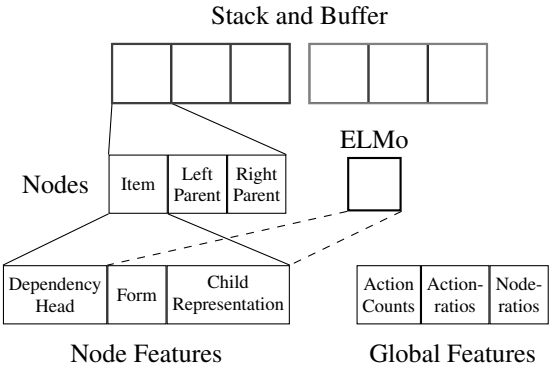

Figure 1: Illustration of the features used by Tüpa. The final feature vector results from the concatenation of all stack and buffer features with the global features. Features dropped after preliminary experiments are omitted for brevity.

4 System

Our system is an ensemble of small feed-forward neural networks. We use three global features: typed absolute counts for previous parser actions and action- and node-ratios (Hershcovich et al., 2017). We further follow the standard in transition-based parsing and extract a set of features based on the top three items on stack and buffer. To capture some of the structure of the partially built graph, we extract the rightmost and leftmost parents and children of the respective items, following Hershcovich et al. (2017). Each of these items is represented by the ELMo embedding of its form, the embedding of its dependency head and the embeddings of all terminal children. We use the average over all ELMo layers to retrieve the embedding of a word. Non-terminal nodes do not have a form or dependency head, hence these are represented by a learned non-terminal embedding. Both the non-terminals and terminals have a third feature, a representation of their children. In the case of terminals, this feature is equal to its form feature. For the non-terminals, it is an aggregation of all its children, produced by the child representation module. Figure 1 illustrates the set of features used by our system. We experimented with richer feature sets, including the last parser actions, named-entity, part-of-speech and dependency types, but dropped them after performing preliminary experiments. The input to the feed-forward module is the concatenation of all features with the output of the child representation module. The classification portion of the system was imple-

mented using Tensorflow (Abadi et al., 2015).

4.1 Representing Non-Terminals

The child representation module aims to enrich the representation of non-terminal nodes. Our initial representation for non-terminal nodes was a set of discrete features describing the number of typed in- and outgoing edges and the nodes' height in the tree. While this might be informative on an abstract level, it does not provide any information about the content covered by this node. We solve this poverty of information by concatenating each of the embeddings of the terminal children of a node with an embedding for the first edge type leading to them. The resulting combination is fed through a dense layer with d neurons, resulting in n vectors with d dimensions where n is the number of terminals under the node. We then reduce the n vectors into a single d dimensional vector by taking the maximum value of each dimension. Figure 2 depicts how the representation of a non-terminal node is obtained. While it would be desirable to process the children using context-aware methods, such as RNNs or self attention, it is not feasible since some of the nodes can have more than 100 children. Future work should explore recursive formulations for representing a node by its direct children instead of relying on all terminal children, performing largely redundant operations for higher nodes.

4.2 Hyperparameters

We apply dropout (Srivastava et al., 2014) with a keep probability of 0.8 to the inputs of all layers. The child processing module is a single layer feed-forward network with 256 hidden units. The feed-forward module is single layer feed-forward network with 512 hidden units. Both modules use the ReLU activation function. Training is performed with the Adam optimizer (Kingma and Ba, 2014) using an initial learning rate of $8\mathrm{e}{-5}$ that is halved every two epochs without an improvement on development accuracy. We halt the training after five epochs without an improvement on development transition accuracy. The models were first trained on the concatenation of the silver and gold data and following the early stopping another time only on the gold data using the same parameters. We use mini-batches of size 192 and evaluate on the development set every 1000 mini-batches. As training time imposes a serious limitation, we did not perform an extensive hyperparameter search and settled on these after initial experiments.

	DAG-F1	Primary F1	Remote F1	Tupa-DAG
English Wiki	0.735	0.741	0.425	0.735
English 20k	0.709	0.719	0.296	0.684
German 20k	0.781	0.788	0.408	0.791
French 20k	0.456	0.464	0	0.487

Table 1: DAG-F1, primary F1 and remote F1 scores with the DAG-F1 score of the baseline on the test sets in the open tracks.

5 Results and Discussion

Table 1 shows the submission scores of our parser trained using the hyperparameters described in Section 4.2 on the test datasets in the open tracks. Since only 15 French passages were available for training, our French results were obtained by first training a model on the concatenation of the French passages and the German 20k training dataset using French ELMo embeddings. After convergence, it was fine-tuned on only the French passages for two epochs. However, this did not provide a significant increase in F1 score over a model trained exclusively on the French passages. All results were produced using a five model ensemble, consisting of the model with the best transition accuracy and the four following it before early stopping. The results show that our parser achieves competitive performance to the baseline while relying on fewer features. In particular, for the English in-domain data, we achieve the same performance as the baseline, for the out-of-domain data we surpass it by 0.025 DAG-F1. In German and French where only in-domain data exists our approach is outperformed by the baseline which we partially attribute to issues in the creation of the silver data. Post-submission results obtained after performing a more exhaustive hyperparameter search on the development set and with correct silver-data surpass the baseline performance on the test sets in all open settings.

6 Further Experiments

In this section, we will describe the findings of our post-evaluation experiments. We evaluated the effect of silver data and provide results for French with silver data. We further performed experiments on non-terminal representations and investigated the effect of model size. Since this only covers a fraction of our experiments and describing them all would be out of scope, we

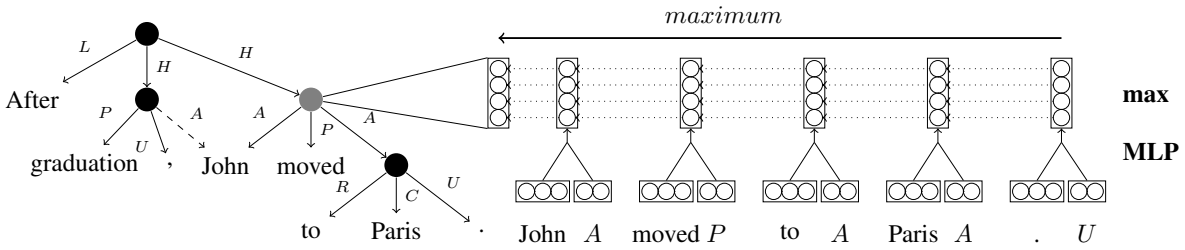

Figure 2: Depiction of a non-terminal representation. The terminal children dominated by the grey node are concatenated with the first edge leading to them and fed through a fully-connected layer. The multiple resulting vectors are reduced into a single one by taking the maximum value of each dimension.

	Full	Submission	Forms only
Gold	0.724	0.733	0.679
Silver+Gold	**0.739**	**0.744**	**0.688**

Table 2: DAG F1 scores on the English development set after training with gold and gold+silver data. Silver data provides a boost for all combinations.

Data	avg. F1	remote F1
Gold	0.456	0.0
Silver+Gold	**0.557**	**0.025**

Table 3: DAG F1 Scores on the French test set with and without silver data. Here in the low-resource setting, the effect of additional data is the largest. Without silver data, the parser did not predict any remote edges correctly.

provide the full results alongside their hyperparameters at https://twuebi.github.io/publications/ucca_post_eval.pdf.

6.1 Silver Data

We measured the effect of silver data on English and French by evaluating several model configurations in two settings. The first setting matches the training data used for the submission and is the concatenation of the gold and silver data. In the second setting, the only available data is the gold data.

English: We trained three models for English. The first model configuration uses all features and corresponds to the model described in the end of Section 4, the second is our submission, described in Section 4. The last model uses only embeddings of the forms and dependency heads. As shown in Table 2, additional training data provides a consistent boost in F1 score across all tested feature combinations. Moreover, it seems that there is a larger effect of the silver data on models with more features, indicating a better estimation of the feature representations based on the additional data.

Low resource setting: Table 3 demonstrates the effect of silver data on French for the submission model configuration. The effect of additional data is the largest in the low-resource setting, providing a boost of 0.1 in average F1 score. Adding the silver data also leads to some of the remote edges being correct, whereas there are no correct remote edges for the gold-only model.

	Full	Submission
Discrete	0.723	0.688
Aggregated	**0.739**	**0.744**

Table 4: Effect of discrete and aggregated non-terminal representations on the DAG F1 score on the English development set. The aggregated representation provides a clear advantage over the discrete one.

6.2 Non-Terminal Representation

To measure the effectiveness of our non-terminal representation, we ran two experiments using silver and gold data. In both cases, we trained one model with aggregated non-terminal representations and one with the discrete representations of typed in- and outgoing edges and the nodes' heights in the tree. The first experiment used all available features. The second was trained with the features of our submission. Table 4 presents the results of the experiments. The explicit child representations provide a clear improvement over the discrete representation. In the second experiment, where no in- and outgoing edges were used and the only non-terminal representations are the left- and rightmost children, the gap increased even further, in fact it is the worst F1 score of all models trained on silver data.

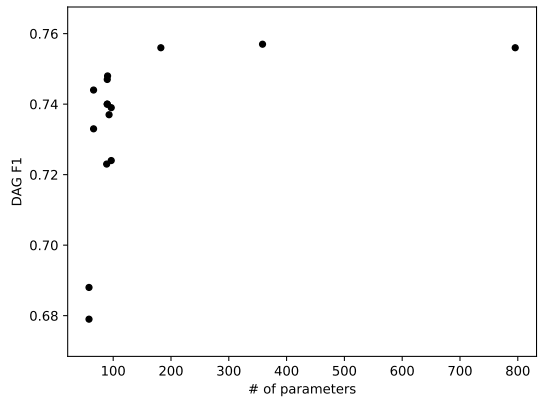

Figure 3: DAG F1 scores on English development data on the y-axis. Million parameters in the models on the x-axis. Larger models seem provide some improvements that begin to level off for big models.

6.3 Bigger means better?

Figure 3 contrasts the number of trainable parameters of the models in our experiments with the F1 score on the English development set. While there are some improvements for larger models, it can be seen that the effect begins to level off at 200M parameters and eventually leads to a small regression with the largest model. Possible causes are overfitting and a lack of training data. Future work should explore whether additional training data allows for larger models. Additional regularization such as L2 regularization might also prove useful. For our experiments, this was out of scope since training so many models was not feasible.

7 Conclusion

In this work, we presented a parser for the semantic grammar formalism UCCA. Our parser relies on a small set of features and achieves competitive performance on the English and German data, but lags behind on French where almost no training data is available. We demonstrated, using ablation experiments, that the explicit representation of non-terminals and additional silver data are crucial for our result. We have further shown that silver data is especially helpful in the low-resource setting where it boosts the average F1 score from 0.456 to 0.557. Future work should investigate how much more improvement additional data can provide. This should be explored both in form of other formalisms (Hershcovich et al., 2018a) and silver data (van Noord et al., 2018). Besides the data aspect, we also believe that improving the non-terminal representation will lead to significant gains. The goal should be to find a representation that leverages the recursive structure of the partially built graph.

Acknowledgements

We would like to thank Çağrı Çöltekin for his extensive comments on an earlier version of this work.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2015. Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

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)*, volume 1, pages 228–238.

Alexandra Birch, Barry Haddow, Ondrej Bojar, and Omri Abend. 2016. HUME: human UCCA-based evaluation of machine translation. *CoRR*, abs/1607.00030.

Johan Bos. 2005. Towards wide-coverage semantic interpretation.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 271–276, Gothenburg, Sweden. Association for Computational Linguistics.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proc. of ACL*, pages 1127–1138.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018a. Multitask parsing across semantic representations. In *Proc. of ACL*, pages 373–385.

Daniel Hershcovich, Leshem Choshen, Elior Sulem, Zohar Aizenbud, Ari Rappoport, and Omri Abend. 2018b. Semeval 2019 shared task: Cross-lingual semantic parsing with ucca-call for participation. *arXiv preprint arXiv:1805.12386*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2016. ccg2lambda: A compositional semantics system. In *Proceedings of ACL 2016 System Demonstrations*, pages 85–90, Berlin, Germany. Association for Computational Linguistics.

Richard T McCoy and Tal Linzen. 2019. Non-entailed subsequences as a challenge for natural language inference. *Proceedings of the Society for Computation in Linguistics*, 2(1):358–360.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 23–28.

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018. Exploring neural methods for parsing discourse representation structures. *arXiv preprint arXiv:1810.12579*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.