Project-then-Transfer: Effective Two-stage Cross-lingual Transfer for Semantic Dependency Parsing

Hiroaki Ozaki, Gaku Morio, Terufumi Morishita, Toshinori Miyoshi
Research and Development Group, Hitachi, Ltd., Japan
{hiroaki.ozaki.yu, gaku.morio.vn, terufumi.morishita.wp, toshinori.miyoshi.pd}@hitachi.com

Abstract

This paper describes the first report on cross-lingual transfer for semantic dependency parsing. We present the insight that there are two different kinds of cross-linguality, namely surface level and semantic level, and try to capture both kinds of cross-linguality by combining annotation projection and model transfer of pre-trained language models. Our experiments showed that the performance of our graph-based semantic dependency parser almost achieved the approximated upper bound.

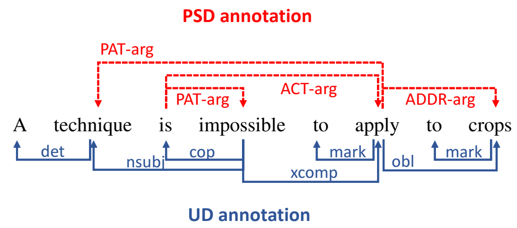


Figure 1: Example dependency annotations. Above: Semantic dependency (PSD). Below: Syntactic dependency (UD). Semantic dependency focuses more on meaning relationship between words.

1 Introduction

Cross-lingual dependency parsing attracted much attention for its powerful representational capability in grammatical and semantic lexical relations (Zhang and Barzilay, 2015; Guo et al., 2015; Ammar et al., 2016; Zeman et al., 2017, 2018; de Lhoneux et al., 2018; Schuster et al., 2019). Several remarkable contributions have been made in *syntactic* dependency parsing, especially on universal dependencies (UD; Nivre et al. 2016). For example, Kondratyuk and Straka (2019) showed that a single multilingually fine-tuned neural model utilizing a pre-trained language model could successfully parse 75 languages in UD with comparable performances to state-of-the-art parsers.

However, cross-lingual *semantic* dependency parsing (Oepen et al., 2014, 2015, 2016), which is totally different dependency structure from syntactic dependencies (shown in Figure 1), has not been explored as far as we know. A reason for this is the lack of parallel graphbanks that cover many languages with consistent annotation policies. One exception is Prague Semantic Dependencies (PSD; Mikulová 2009), which is a treebank of bi-lexical semantic graphs and contains over 30,000 pairs of parallel annotated sentences from the Wall Street Journal in English and Czech.

Considering these circumstances, we propose to train semantic dependency parsers by capturing commonalities across languages as a remedy for the absence of massive multilingual graphbanks. Our work draws on the intuition that cross-linguality exists in both superficial level and semantic level. Accordingly, we leverage a two-stage fashion involving treebank-based transfer and model-based transfer.

Treebank-based transfer, often called annotation projection, is a method of projecting source language annotations to a target language by using a mapping function such as word alignment. The annotation projection has been reported as a promising approach under truly low-resource settings for UD parsing (Rosa and Mareček, 2018). However, annotation projection often suffers from noise in word alignment (Damonte and Cohen, 2018). For the model-based transfer, several studies on transferring contextualized word vectors have reported that it improves the parsing performance (Mulcaire et al., 2019; Kondratyuk and Straka, 2019).

Our experiments on PSD graphbank indicate that the optimal performance can be achieved by incorporating the two-stage transfer. Surprisingly, we observed improvement even when the projected treebank was erroneous. Furthermore, the two-

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 2586–2594
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

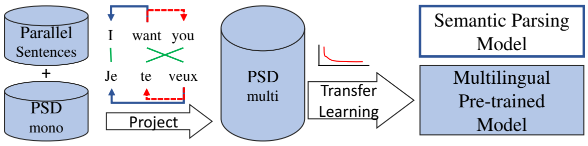


Figure 2: Project-then-Transfer approach.

stage transfer method achieved almost upper-bound performance, which was approximated by evaluating the cross-linguality of PSD annotation through the projection. We also provide detailed analyses from both perspectives of cross-linguality.

2 Related Work

Semantic Dependency Parsing: The topic of semantic dependency parsing has spurred enduring interest (Peng et al., 2017, 2018; Dozat and Manning, 2018; Wang et al., 2019; Kurita and Søgaard, 2019). Much of the current interest lies in higher-level interactions between relations. In parallel with our study, Aminian et al. (2020) shows improvement of PSD parsing trained on cross-lingually projected graphbank with multitask training of UD parsing as an auxiliary task. Unlike their work, we perform a zero-shot training with UDify pretrained model to validate the hypothesis of the two different cross-linguality.

Utilizing Models across Different Graphbanks: Parsing semantic graphs in different semantic abstraction levels was introduced as CoNLL shared task 2019 (Oepen et al., 2019). Candidate teams tackled this problem with methods such as transition-based parsers (Hershcovich et al., 2018; Bai and Zhao, 2019; Lai et al., 2019) and graph-based parsers (Zhang et al., 2019; Koreeda et al., 2019). Small improvements were reported in both approaches, but improving semantic parsing on different semantic graphs remains a difficult problem.

3 Transfer Strategies

To enable cross-lingual semantic parsing, we focused on two different types of cross-linguality, namely *cross-linguality on surface* and *cross-linguality in semantics*. *Cross-linguality on surface* lies on our hypothesis of typological correspondences among most of languages. For example, annotation projection, which is a treebank-based transfer, assumes *cross-linguality on surface* and projects source language annotations to a target language. On the other hand, *cross-linguality in semantics* is based on the assumption that lexical-, phrase-, or sentence-level meaning correspondences may exist among most of languages. Recently, multilingual BERT (Devlin et al., 2019; Pires et al., 2019) directly captures *cross-linguality in semantics* beyond lexicons by large-scaled language model training on parallel corpora.

Though both annotation projection and multilingual pre-trained model can handle either *cross-linguality on surface* or *in semantics*, we argue that they could not utilize both cross-linguality in effective way. Hence, we propose two-stage transfer which incorporates both methods, to capture the two kinds of cross-linguality as possible. We firstly introduce the two transfer methods for applying them to PSD graphbank, and then we explain our two-stage transfer method; Project-then-Transfer.

Annotation Projection: As aforementioned, this is the approach focuses on *cross-linguality on surface*. We trained word alignment model on PSD graphbank, and then projected all annotations in a monolingual graphbank to the other language.

Zero-shot Model Transfer: In this study, we transferred only pre-trained language models, because we aimed to focus on *cross-linguality in semantics* more.We trained PSD parsers with multilingual pre-trained model on a monolingual graphbank in PSD, and then apply the monolingually trained parsers to the other language.

Project-then-Transfer: We incorporate both transfer methods by applying them in two-stage fashion as shown in Figure 2. Firstly, we prepared multilingually projected PSD graphbanks. We automatically generated PSD annotations on English sentences in a multilingual parallel corpus by the previously introduced English PSD parser which was created in the zero-shot approach. By utilizing bi-lingual word alignment, we projected PSD annotations on English to other languages. We finally trained Project-then-Transfer models on a concatenated graphbank of both original and projected PSD.

4 Experiments

4.1 Setup and Implementations

To perform word alignment, we used an IBM2 aligner *fast_align*[1] (Dyer et al., 2013).

[1]https://github.com/clab/fast_align

Language	Model	Approach	UP	UR	UF	LP	LR	LF	LF/UF
Czech (Trained on English)	Graph-UDify	**Project-then-Transfer (ours)**	86.5	**78.7**	**82.5**	**62.4**	**56.7**	**59.4**	72.0
	Graph-BERT		80.1	62.7	70.4	59.7	46.8	52.5	74.6
	Graph-UDify	Zero-shot	**86.9**	75.5	80.8	61.8	53.7	57.5	71.1
	Graph-BERT		79.1	61.1	69.0	58.7	45.3	51.1	74.1
	fast_align	Projection	49.3	40.7	44.6	37.1	30.6	33.5	**75.1**
English (Trained on Czech)	Graph-UDify	**Project-then-Transfer (ours)**	77.4	**79.7**	**78.5**	57.4	**59.1**	**58.2**	74.1
	Graph-BERT		66.9	57.7	62.0	50.7	43.7	46.9	75.6
	Graph-UDify	Zero-shot	**81.0**	72.4	76.5	**59.2**	52.9	55.9	73.1
	Graph-BERT		73.6	58.1	65.0	56.5	44.6	49.9	**76.8**
	Transition-BERT		55.2	22.6	32.3	41.8	17.2	24.4	75.5
	fast_align	Projection	48.6	44.3	46.3	35.6	32.4	33.9	73.2
Li et al. (2019) (English monolingual training)			93.	92.	92.5	82.	81.	81.7	88.3

Table 1: SDP scores for each model and approach. U and L stand for "unlabeled" and"labeled" respectively. P, R, and F stand for "precision", "recall" and "F1-score" respectively. LF/UF is a proxy metric of label prediction accuracy. Bold values represent the best scores. Li et al. (2019) is the best PSD parser at CoNLL 2019 shared task.

To perform model-based transfer, we used mainly graph-based parsers, but we also used a transition-based parser for a comparison purpose. Our graph-based PSD parser employed UDify architecture[2] (Kondratyuk and Straka, 2019). We replaced activation function of biaffine attention layers in UDify with sigmoid activation (Dozat and Manning, 2018). We trained two variances of graph-based parsers, and a transition-based parser:

Graph-BERT: We trained it with mulitilingual BERT as Kondratyuk and Straka (2019) did.

Graph-UDify: We trained it with UDify's pre-trained language model[3] instead of multilingual BERT. Since UDify is pre-trained on many languages in UD, we expect that it capture more *cross-linguality on surface* than BERT.

Transition-BERT: We used an architecture introduced by Che et al. (2019)[4], which was the best transition-based parser in the CoNLL 2019 shared task (Oepen et al., 2019). We trained it from scratch with the same hyperparameters given by the source code.

The pre-trained multilingual BERT[5] was downloaded via the above parser implementations. We used mtool[6] to evaluate SDP scores (Oepen et al., 2014) as metrics for parsing performance. A list of the best hyperparameters is available in Appendix. We added "tag_loss_w", which is a constant multiplied by the loss of relation label predictions.

We divided PSD graphbank into three splits, namely train-set (30,000 pairs), dev-set (2000 pairs), and test-set (3653 pairs). We selected the best models by monitoring the labeled F1-score of SDP on the dev-set of the target language and evaluated the scores on the test-set of the target language. We chose Parallel Universal Dependencies (PUD; Zeman et al. 2017) as additional multilingual parallel corpora for Project-then-Transfer, because they contain 1,000 parallel sentences for 18 languages, with mostly consistent UD annotations. Further details are in Appendix.

4.2 Results and Discussion

Table 1 shows the SDP scores for each model in each approach. Firstly, we focus on the cross-linguality of PSD annotations by the annotation projection. Unlabeled scores of projection models were within a range of 0.4 - 0.5. Since alignment error rate (AER) of English-Czech reported around 0.25 (Legrand et al., 2016), edge projection accuracy could be estimated as $(1 - \text{AER})^2 \approx 0.56$ [7]. Annotation agreement rate of relations[8] between the two languages was estimated to fluctuate between 0.7 to 0.9 according to the mitigation efficacy of alignment error. Annotation agreement rate of relation labels was estimated as about 0.75 by comparing unlabeled and labeled scores (LF/UF of fast_align model). These rates could be upper bounds of performances.

By comparing monolingual training (Li et al.,

[2]https://github.com/Hyperparticle/UDify (Pre-trained models are also available from the link.)

[3]We did not utilize biaffine and MLP layers of UDify.

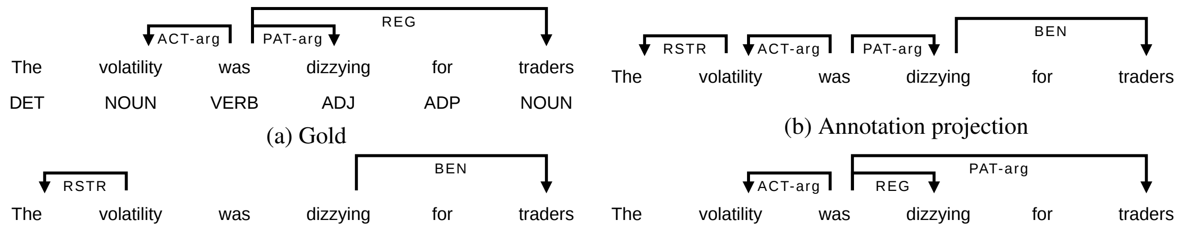
[4]https://github.com/DreamerDeo/HIT-SCIR-CoNLL2019

[5]https://github.com/google-research/bert/blob/master/multilingual.md

[6]https://github.com/cfmrp/mtool

[7]Suppose one edge has two nodes A and B, then edge projection accuracy is estimated as probability that both projected nodes A' and B' are correct.

[8]We simply divided the unlabeled scores by the projection accuracy estimated by AER.



(a) Gold (b) Annotation projection (c) Graph-UDify zero-shot (d) Graph-UDify Project-then-Transfer

Figure 3: Example gold and Graph-UDify outputs in each scenario (English).

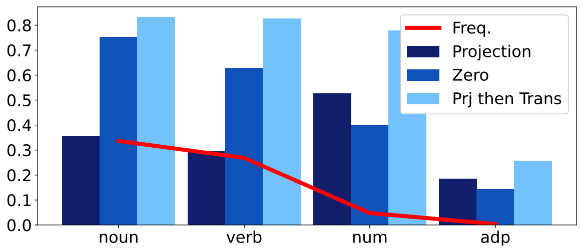


Figure 4: Relation accuracy for each UPOS type.

Model	language	UF	LF
Graph-BERT + PUD_Czech	en2cs	68.9	50.8
Graph-UDify + PUD_Czech	en2cs	80.6	57.8

Table 2: Unlabeled and labeled scores trained on English PSD with projected PUD_Czech.

2019), unlabeled-F of Project-then-Transfer (UDify) was about 85% of that of monolingual training. This ratio is in the range of estimated annotation agreement rate of relations.

Models and Approaches Comparison: As we can see from Table 1, the graph-based models outperformed the Transition-BERT, especially, graph-based UDify models demonstrated superiority to the other models. Graph-UDify of Project-then-Transfer approach, which is the best model, achieved an unlabeled F1-score of 82.5, that is close to the upper-bounds estimated above. In additions, there were few differences in LF/UF scores, which are also close to the upper-bound of relation label prediction. Thus, our best model achieved high performance, which is close to theoretical upper bounds. This indicates that bi-lexical relations captured by syntactic dependency are also helpful for parsing semantic dependency, yet there remaining information that were not captured in the UDify. We claim that the missing information was related to *cross-linguality on surface*, then we perform a deeper analysis on this in the following paragraph.

What is NOT captured by Pre-trained Models?: Figure 3 shows examples of gold and Graph-UDify (cs2en) outputs. The annotation projection had managed completely project *unlabeled* relations in the source language, but a swapping had happened between two relations, namely "REG" and "PAT-arg", which had been caused by alignment errors.

We observed that parsers based on the model-based transfer often failed to parse relations which contain functional words. This phenomenon can be observed in Figure 3c. Those relations containing functional words tended to be successfully converted by the annotation projection. Hence, we obtained better results with the Project-then-Transfer approach as shown in Figure 3d. This implies that pre-trained models including UDify represent rather *semantic* bi-lexical relations than grammatical ones.

We performed a further analysis on cross-linguality of UDify model. Figure 4 shows relation accuracy for each of four UPOS, namely *noun*, *verb*, *num* and *adp*. We calculated conditional "unlabeled" relation accuracy, which measures whether source or target word belongs to the specific UPOS type. By focusing on the accuracy of *num* (numeric) and *adp* (adposition), which are considered to be hard-to-contextualize examples, the annotation projection outperformed the zero-shot approach. The Project-then-Transfer approach improved the accuracy for almost all UPOS types including *num* and *adp*.

Cross-linguality on Surface: Table 5 shows SDP scores trained on English PSD with projected PUD_Czech. The performances were comparable to the zero-shot approach, but less than those of the Project-then-Transfer approach. Hence, multilingually projected treebank is significant to improve the performances. This implies that *cross-linguality on surface* can be captured by training on multilingually projected treebank.

5 Conclusion

This paper described transfer methods for cross-lingual semantic dependency parsing. We showed that both *cross-linguality on surface* and *in semantics* were necessary to improve the performance. Consequently, we achieved almost the upper bound performance approximated by the annotation projection. The results encouraged us to develop cross-lingual semantic dependency parser for many languages. We will further conduct explore these models, and evaluations on cross-linguality across languages broadly.

Acknowledgments

Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by the National Institute of Advanced Industrial Science and Technology (AIST) was used. We would like to thank the anonymous reviewers for their helpful comments. We also thank Dr. Masaaki Shimizu for the convenience of the computational resources.

References

Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. 2020. Multitask learning for cross-lingual transfer of broad-coverage semantic dependencies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8268–8274, Online. Association for Computational Linguistics.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Hongxiao Bai and Hai Zhao. 2019. SJTU at MRP 2019: A transition-based multi-task parser for cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 86–94, Hong Kong. Association for Computational Linguistics.

Wanxiang Che, Longxu Dou, Yang Xu, Yuxuan Wang, Yijia Liu, and Ting Liu. 2019. HIT-SCIR at MRP 2019: A unified pipeline for meaning representation parsing via efficient training and effective encoding. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 76–85, Hong Kong. Association for Computational Linguistics.

Marco Damonte and Shay B. Cohen. 2018. Cross-lingual abstract meaning representation parsing. 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 1146–1155, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but More Accurate Semantic Dependency Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1234–1244, Beijing, China. Association for Computational Linguistics.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018. Multitask parsing across semantic representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. 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 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Yuta Koreeda, Gaku Morio, Terufumi Morishita, Hiroaki Ozaki, and Kohsuke Yanai. 2019. Hitachi at MRP 2019: Unified encoder-to-biaffine network for cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 114–126, Hong Kong. Association for Computational Linguistics.

Shuhei Kurita and Anders Søgaard. 2019. Multi-task semantic dependency parsing with policy gradient for learning easy-first strategies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2420–2430, Florence, Italy. Association for Computational Linguistics.

Sunny Lai, Chun Hei Lo, Kwong Sak Leung, and Yee Leung. 2019. CUHK at MRP 2019: Transition-based parser with cross-framework variable-arity resolve action. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 104–113, Hong Kong. Association for Computational Linguistics.

Joël Legrand, Michael Auli, and Ronan Collobert. 2016. Neural network-based word alignment through score aggregation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 66–73, Berlin, Germany. Association for Computational Linguistics.

Miryam de Lhoneux, Johannes Bjerva, Isabelle Augenstein, and Anders Søgaard. 2018. Parameter sharing between dependency parsers for related languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4992–4997, Brussels, Belgium. Association for Computational Linguistics.

Zuchao Li, Hai Zhao, Zhuosheng Zhang, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2019. SJTU-NICT at MRP 2019: Multi-task learning for end-to-end uniform semantic graph parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 45–54, Hong Kong. Association for Computational Linguistics.

Marie Mikulová. 2009. Pokyny k překladu určené překladatelům, revizorům a korektorům textů z wall street journal pro projekt PCEDT. Technical Report TR-2009-41, ÚFAL MFF UK.

Phoebe Mulcaire, Jungo Kasai, and Noah A. Smith. 2019. Low-resource parsing with crosslingual contextualized representations. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 304–315, Hong Kong, China. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. 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 2016)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Stephan Oepen, Omri Abend, Jan Hajic, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdenka Uresova. 2019. MRP 2019: Cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 1–27, Hong Kong. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Zdeňka Urešová. 2016. Towards comparability of linguistic graph Banks for semantic parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3991–3995, Portorož, Slovenia. European Language Resources Association (ELRA).

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. 2015. SemEval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926, Denver, Colorado. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72, Dublin, Ireland. Association for Computational Linguistics.

Hao Peng, Sam Thomson, and Noah A. Smith. 2017. Deep multitask learning for semantic dependency parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2037–2048, Vancouver, Canada. Association for Computational Linguistics.

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith. 2018. Learning joint semantic parsers from disjoint data. 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 1492–1502, New Orleans, Louisiana. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Rudolf Rosa and David Mareček. 2018. CUNI x-ling: Parsing under-resourced languages in CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018*

Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pages 187–196, Brussels, Belgium. Association for Computational Linguistics.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. 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 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinyu Wang, Jingxian Huang, and Kewei Tu. 2019. Second-order semantic dependency parsing with end-to-end neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4618, Florence, Italy. Association for Computational Linguistics.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Yuan Zhang and Regina Barzilay. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1857–1867, Lisbon, Portugal. Association for Computational Linguistics.

Model	Parameters
fast_align	– iteration: 10 – num. of trials: 1 – average runtime: less than 1min.
Graph-UDify	– batch size: 32 – learning rate: 1e-3 – activation: relu – beta: [0.99, 0.99] – tag_representation_dim: 128 – tag_loss_w: 0.1 – num. of trials: 32 for each – average runtime: 12 hrs. – num. of params.: 225695576
Transition-BERT	– batch size: 4 – num. of trials: 1 – average runtime: 48 hrs. – num. of params.: 201555396

Table 3: The best hyperparameters and training settings for fast_align, Graph-UDify, and Transition-BERT.

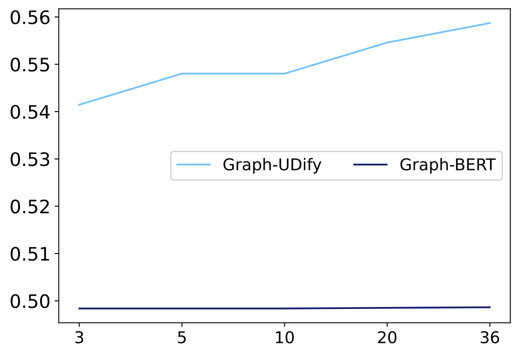


Figure 5: Expected Validation Performances.

Yue Zhang, Wei Jiang, Qingrong Xia, Junjie Cao, Rui Wang, Zhenghua Li, and Min Zhang. 2019. SUDA-Alibaba at MRP 2019: Graph-based models with BERT. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 149–157, Hong Kong. Association for Computational Linguistics.

A Appendices

In this section, we provide details of training setup, analyses on relation accuracy and multi-task learning results.

A.1 Detail of Training Setups

In this sub-section, we provide training setups, which are for the reproducibility criteria. Table 3 shows all detailed settings. We did not performed a severe automatic hyperparameter tuning, but did a manual tuning. Thus our best hyperparameters may be different from the true best hyperparam-

Language	Model	Approach	UP	UR	UF	LP	LR	LF	LF/UF
Czech	Graph-BERT	**Project-then-Transfer**	80.5	62.0	70.1	60.1	46.3	52.3	74.6
	Graph-UDify		86.4	78.3	82.7	62.5	57.3	59.8	72.3
(Trained on English)	Graph-BERT	Zero-shot	78.8	60.3	68.4	58.4	44.7	50.6	74.0
	Graph-UDify		86.5	75.6	80.7	61.3	53.6	57.1	70.8
	fast_align	Projection	49.3	40.7	44.6	37.1	30.6	33.5	75.1
English	Graph-BERT	**Project-then-Transfer**	67.3	57.4	62.0	49.2	42.0	45.3	73.1
	Graph-UDify		77.3	79.6	78.5	55.7	57.4	56.5	72.0
(Trained on Czech)	Graph-BERT	Zero-shot	73.4	57.6	64.5	54.8	43.0	48.2	74.7
	Graph-UDify		81.3	72.5	76.7	57.6	51.3	54.3	70.8
	Transition-BERT		55.4	62.6	58.8	40.4	45.7	42.8	72.8
	fast_align	Projection	48.6	44.3	46.3	35.6	32.4	33.9	73.2

Table 4: SDP scores on the dev-set for each model and approach.

Model	language	UF	LF
Graph-BERT + PUD_Czech	en2cs	69.0	50.7
Graph-UDify + PUD_Czech	en2cs	80.9	58.0

Table 5: Unlabeled and labeled dev-set scores of models trained on English PSD with projected PUD_Czech.

eters. We tuned hyperparamters in zero-shot approach, then we reused the best hyperparameters in Project-then-Transfer approach. We trained all models with NVIDIA V100 on Ubuntu 18.04. Our GPU environment is a mixture of both 32GB and 64GB memories.

We obtained PSD treebank from the Linguistic Data Consortium (LDC) [9]. We converted original SDP format data to MRP format before the training. This SDP to MRP graph conversion is a loss-less conversion.

A.2 Performances on Dev-set

Figure 5 shows *expected validation performances*. Table 4 and Table 5 show performances on the dev-set. Most of scores were consistent with the performances on the test-set. Only transition-based model is an exceptional case. Though over-fitting seemed to happen, its performances are still lower than those of graph-based models.

A.3 Full Results of Relation Accuracy

Figure 6 shows relation accuracy for all UPOS types. We can see that zero-shot performances of eleven types, namely *noun*, *verb*, *propn* (proper noun), *conj* (conjunction), *pron* (pronoun), *adv* (adverb), *punct* (punctuation), *det* (determiner), *part* (particle), *cconj* (coordinating conjunction), and *x* (other), are outperformed those of annotation projection, and all content words are included in this group. Zero-shot performances of the other five types, *num*, sym (symbol), *adp*, *sconj* and *intj* (interjection), are comparable or underwhelmed to those of annotation projection. Because the words categorized as *intj* only appeared a few times in this analysis, we could not make a discussion regarding *intj*.

A.4 Multi-Task Learning

We argue that it is natural to perform multi-task learning (MTL) of UD and PSD dependencies when both annotations are available, since UDify's pre-train model, which is trained on UD annotations, improved the performances. Firstly, we added UD annotation on PSD treebank by existing UD parser UDPipe[10]. Our MTL setting is to share only BERT layers, but higher layers including scalar-mix layers are distinct. We used UDify "as is" for UD prediction. A Loss function to perform MTL is a simple linear combination of that of UDify and our PSD model. We show the MTL results in Table 6. Performances were degraded by comparing to those of non-MTL models. This could be because UD and PSD annotations are contradictive to perform MTL.

[9] https://catalog.ldc.upenn.edu/LDC2016T10

[10] http://ufal.mff.cuni.cz/udpipe

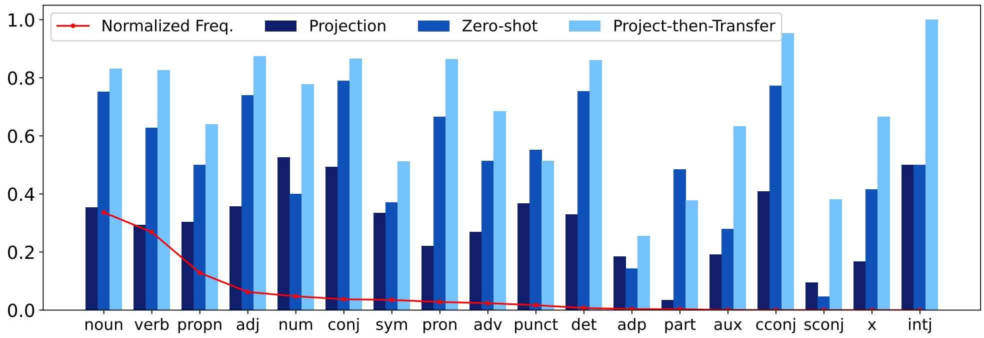


Figure 6: Relation accuracy for all UPOS types.

Dataset	Model	Approach	UP	UR	UF	LP	LR	LF	LF/UF
test-set	Graph-BERT-MTL	Project-then-Transfer	83.7	60.5	70.2	61.6	44.5	51.7	73.6
	Graph-UDify-MTL		86.4	73.9	79.7	63.3	54.1	58.4	73.3
dev-set	Graph-BERT-MTL	Project-then-Transfer	83.9	57.0	67.9	61.9	42.0	50.0	73.6
	Graph-UDify-MTL		85.9	73.5	79.2	63.3	54.1	58.3	73.6

Table 6: SDP scores for MTL model and approach.