Scalable Cross-lingual Treebank Synthesis for Improved Production Dependency Parsers

Yousef El-Kurdi, Hiroshi Kanayama, Efsun Sarioglu Kayi[*],
Todd Ward, Vittorio Castelli, Hans Florian
IBM Research
yousefelk@us.ibm.com, hkana@jp.ibm.com, efsun@gwu.edu
toddward@us.ibm.com, vittorio@us.ibm.com, raduf@us.ibm.com

Abstract

We present scalable Universal Dependency (UD) treebank synthesis techniques that exploit advances in language representation modeling which leverage vast amounts of unlabeled general-purpose multilingual text. We introduce a data augmentation technique that uses synthetic treebanks to improve production-grade parsers. The synthetic treebanks are generated using a state-of-the-art biaffine parser adapted with pretrained Transformer models, such as Multilingual BERT (M-BERT). The new parser improves LAS by up to two points on seven languages. The production models' LAS performance improves as the augmented treebanks scale in size, surpassing performance of production models trained on originally annotated UD treebanks.

1 Introduction

Dependency parsers are important components in many NLP systems, such as language understanding, semantic role labeling and relation extraction (Marcheggiani and Titov, 2017; Zhang et al., 2018). Universal Dependencies (UD) (Nivre et al., 2020; Zeman et al., 2018) are becoming a widely accepted standard among many NLP practitioners for definition of syntactic structures and treebanks. However, production parsers require custom tokenization policies and Part of Speech (PoS) tagging, mostly dictated by supported downstream applications. In addition, parsers in production environments require fine balancing of demands for model accuracy, service performance, response time and constraints on hardware resources, making the design of an industrial-grade parser a challenge. Hereby, we introduce data augmentation techniques to improve production parsers without violating their architectural constraints.

Since their early inception, advances in language representation modeling lead to major improvements in many NLP tasks (Wang et al., 2018; Moon et al., 2019). Representations trained on various language modeling objectives, ranging from context free embeddings (Pennington et al., 2014; Mikolov et al., 2013), to deep context aware representations (Peters et al., 2018; Le and Mikolov, 2014; Devlin et al., 2018), were trained on massive amounts of unlabeled multilingual text, greatly enabling transfer learning opportunities for NLP tasks. Particularly, models such as BERT (Devlin et al., 2018), ALBERT (Lan et al., 2020), RoBERTa (Liu et al., 2019) and XLM (Lample and Conneau, 2019) employ a masked language modeling objective (Taylor, 1953) on a bidirectional self-attention encoder (Vaswani et al., 2017) enabling such models to utilize both left and right context for each word representation. Pretrained multilingual BERT (M-BERT) was used for dependency parsing in (Kondratyuk and Straka, 2019) aiming to create a single multilingual model. This work, in contrast, shows that parsing performance for a particular language can considerably be improved when adapting the biaffine-attention parser (Qi et al., 2018) with a selected set of pretrained Transformer models while training on multilingual subsets of selected language family treebanks. We then use this novel parser to project synthetic treebanks, which are used in a teacher-student technique to improve the accuracy of a fast production parser.

Our approach can generally be described as a form of model compression which was introduced by (Bucilu et al., 2006), and later reformulated and generalized as neural network knowledge distillation by

[*] Work done during AI Residency at IBM Research.

Proceedings of the 28th International Conference on Computational Linguistics: Industry Track, pages 172–178
Barcelona, Spain (Online), December 12, 2020

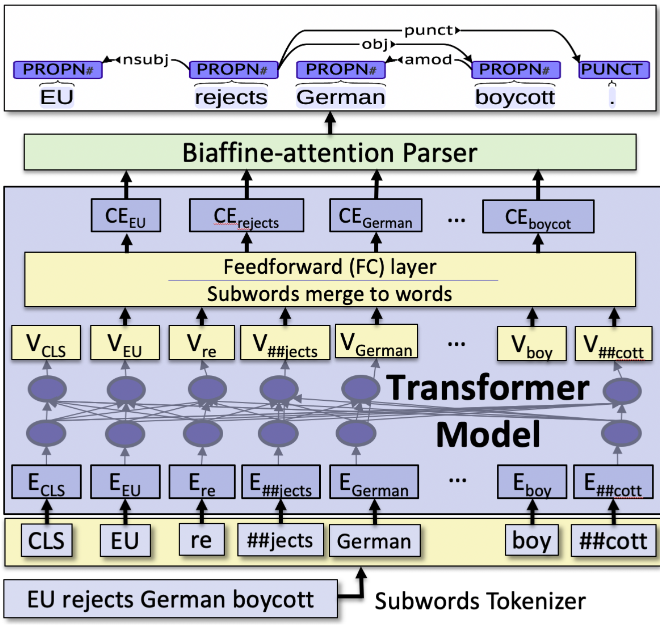

Figure 1: Transformer enhanced biaffine-attention Parser (TBAP).

(Hinton et al., 2015). However, instead of using a large number of ensemble for a teacher model, we use a deep neural network parser augmented with a large transformer-based pretrained model creating a new parser that advances the current state of the art. Since the pretrained transformer model can be trained on large amounts of unlabeled monolingual and multilingual data of various domains and languages, the teacher model gains improved generalization performance that is facilitated by both cross-lingual and cross-domain transfer learning. Our student model is a non-neural net based model that is designed to be fast and efficient in production environments.

Conventionally, parsers are trained on human annotated treebanks, which can be both costly and limited in quantity. Certain languages may not have enough annotation resources, have very small amount of data, or data that carries non-commercial licenses. In other cases, the data may be available in a specific topical domain resulting in models that perform poorly on unrelated domains. In addition, annotation errors can be common in some treebanks. To address these data challenges, we use cross-lingual transfer learning and pretrained deep contextualized representations to create a novel parser that helps generate synthetic data. We describe a production parser trained on these data, whose performance increases as the synthetic data size surpasses that of human annotated data.

2 System Description

The aim of our system is to produce synthetically labeled treebanks in order to significantly improve the accuracy of a production parser. The synthetic data will be generated using a different parser that is higher in quality. We create a new parser using two key components: the deep biaffine-attention parser (Qi et al., 2018) and a pretrained Transformer model. Not only does such a setup improve the parsing accuracy, as shown in Section 3, the incorporation of Transformer models facilitates greater degree of generalization and domain adaptation. In the sections below, we detail the training data augmentation process as well as the new parser architecture.

2.1 Transformer Enhanced Biaffine-Attention Parser

Figure 1 shows the architecture of the Transformer enhanced Biaffine-Attention Parser (TBAP). The Transformer provides contextualized word representations for each input sentence to the BiLSTM layer of the biaffine parser. First, a tokenized input sentence is passed through the Transformer. The Transformer further breaks word tokens into subword tokens. This is done in order to significantly reduce the size of the fixed vocabulary representation in the output prediction layer (Sennrich et al., 2016) overcom-

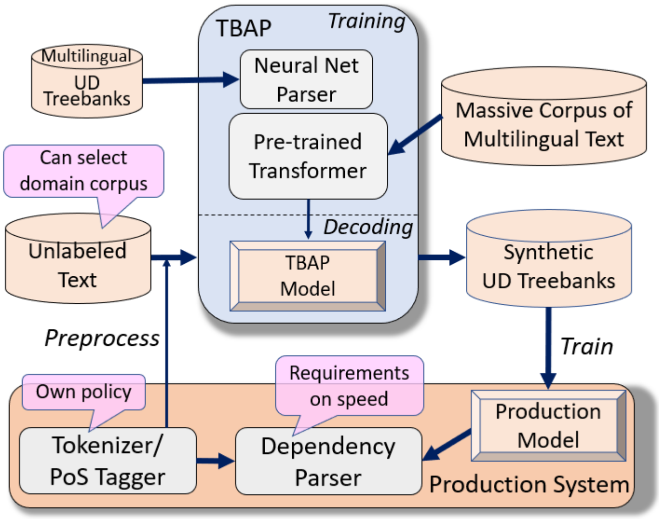

Figure 2: The data augumentation process for training a production parser.

ing the open-vocabulary problem. We then take the sum of the last four encoder layers of the Transformer as the output representation, which is comprised of the contextualized subword representations of the input sentence. Afterwords, two operations are performed on the Transformer's output. Subword token representations (also referred to as WordPiece tokens for BERT) are merged back into word-based representations. Merging the subword representations can either be done by averaging, maxpooling, or simply taking the first subword of each word. A forward Fully Connected (FC) layer is then applied to the merged subword representations, resulting in a sequence of representations aligned for each word of the tokenized input sequence.

The TBAP is trained on available treebanks. This is a process where the Transformer itself is fine-tuned by allowing backpropagation to flow through it during training. Alternatively, freezing the Transformer layers while training can help in speeding up the training process with some drop in performance.

2.2 Data Augmentation Process

Figure 2 outlines the stages of multilingual treebank generation. Initially, the TBAP is trained with available treebanks. Depending on the type of Transformer model used, two training approaches can be followed, monolingual and multilingual. Monolingual training can be applied when monolingual Transformer models are used. Pretrained monolingual Transformer models are available for certain languages, such as English, German, French, Chinese, Japanese as well as others. Performance can particularly be improved for these languages due to both the abundance and specialization of their monolingual data. Multilingual Transformer models, such as Multilingual-BERT (M-BERT) are trained on more than 100 languages. When using M-BERT, both monolingual as well as multilingual treebanks can be used to train the parser. Low resource languages can particularly benefit from cross-lingual transfer learning.

2.3 Fast Production Parser

Our production parser should meet rigid criteria regarding runtime speed; thus, we choose the arc-eager algorithm (Nivre, 2004) trained with features similar to those used by Chen and Manning (2014). To maintain UD compatability for existing downstream tasks, the tokenization and PoS tagging should not be modified even if they do not completely follow the definitions from UD. As shown in Figure 2, the dependency parser takes the tokenizer and PoS tagger's results as input in order to produce UD-based syntactic structures.

UD	Parser	Transformer Model	LAS
en_ewt	SNLP		89.50
	TBAP	BERT-base-en	91.36
		BERT-large-en	**92.38**
		Multilingual-BERT	91.14
		Albert-xxlarge	92.12
		Roberta-large	91.02
		XLM	91.56
de_gsd	SNLP		86.16
	TBAP	BERT-base-de	**87.92**
		Multilingual-BERT	87.35

Table 1: LAS results for monolingual and multilingual Transformer models.

		M-BERT	Multilingual
UD	SNLP	TBAP	Treebanks
nl_alpino	93.76	94.01	de_gsd, en_ewt
fr_gsd	92.13	93.54	
it_isdt	92.61	94.66	es_gsd, pt_bosque
pt_bosque	90.49	91.08	it_isdt, es_gsd
es_gsd	89.94	91.72	it_vit, pt_bosque

Table 2: LAS comparison SNLP and TBAP.

UD	Tags	No-Tags
fr_gsd	72.30	84.90
pt_bosque	62.59	75.47

Table 3: TBAB LAS for unmatched tags.

3 Experiments

In this section, we show results demonstrating the improved performance of the new TBAP architecture on seven languages. We also show the effectiveness of the treebank synthesis technique when used in the augmented training of a production parser.

3.1 Transformer Enhanced Biaffine-Attention Parser (TBAP)

The TBAP is implemented by combining two key components, a pretrained Transformer model and the Biaffine-attention parser. The interface to the pretrained Transformer models was obtained from the Hugging Face's Transformers library (Wolf et al., 2019). The implementation of the biaffine-attention parser was obtained from the open-source StanfordNLP (SNLP) library (Qi et al., 2018). The FC and the subwords merge layers were added between the Transformer and the biaffine parser. We have adapted the dependency parser component to be connected to the pretrained transformers and left other components of the SNLP pipeline unchanged. In fact, since the synthetic data is being preprocessed by the production parser's tokenizer and tagger, we only needed to adapt the UD parser and disable the other modules in the pipeline. Other modifications were performed on the UD parser to make it more suitable for our task such as changing the internal dimensions of the embeddings layers, adjusting the vocabulary data structures to make them suitable for multilingual training, and controlling which UD features can be used when training the UD parser. The PyTorch[1] library is used to implement the TBAP code.

We use the standard UD treebanks v2.6 in our evaluations of the TBAP models. The UD v2.6 designated devset of each treebank is used as a tune-set for early stopping criterion during training. The UD v2.6 testset of each treebank is used for Labeled Attachment Score (LAS) results in the tables below. All models generated from UDs are for evaluation purposes. In most cases, we re-trained the SNLP (unmodified) parser in order to obtain improved baseline scores over the existing pretrained models.

Table 1 shows the LAS results of TBAP with various Tranformer models compared with the baseline SNLP parser. Since we only modified the dependency parser, we compute scores based on gold sentences, tokens and tags. Table 1 shows that TBAP with any of the used Transformer models improves LAS over the baseline parser. Also the best results are observed when using a Transformer model trained monolingually on the corresponding language. This can be attributed to the larger amount of monolingual text used to train the monolingual Transformer. Also in monolingual language models, the subword splitting models are improved, which results in less splitting and consequently improved contextual representations. For English, BERT large outperforms the base one. Table 2 shows LAS results for training on different language UDs using M-BERT TBAP. M-BERT TBAP consistently outperforms the baseline SNLP parser.

In the typical case where the synthetic data will be used to train a different production parser, it will be first preprocessed by the production parser; that is sentence segmented, tokenized, and PoS tagged by the

[1] `https://pytorch.org`

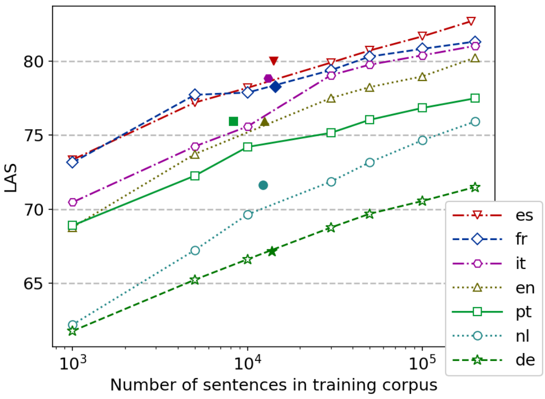

Figure 3: LAS against the size of synthetic training corpora. The filled symbols (*e.g.* ▲, ★) denote the results with the corresponding UD corpora.

UD	UD Corpus		Synthetic	
	Size	**LAS**	**Size**	**LAS**
es_gsd	14.2k	80.0	190k	82.7
fr_gsd	14.4k	78.3	200k	81.3
it_isdt	13.0k	78.8	300k	81.6
en_ewt	12.5k	75.9	200k	80.2
pt_bosque	8.3k	75.9	300k	77.5
nl_alpino	12.3k	71.7	300k	76.7
de_gsd	13.8k	67.2	200k	71.5

Table 4: Results of the production parser for seven languages on UD 2.6 testsets. Comparing training data by number of sentences and LAS (F1) for both the original UD and the larger synthetic corpora.

production parser's own pipeline. This preprocessing is required so that the parser's output is compatible with other downstream NLP tasks. This means that the preprocessing will not necessarily be consistent with the treebank from its corresponding language. In order to improve the robustness of the synthetic data under different preprocessing requirements, the M-BERT TBAP must be trained without relying on such predicted tags. Table 3 shows the effect of removing the tags while training the M-BERT TBAP for synthetic data generation. As expected the overall LAS consequently drops; however, the no-tags model's scores shows less of an impact for the preprocessed testset.

3.2 Augmented Training of the Production Parser

We retrained the production parser using the synthetic data generated by the methods above. Table 4 shows the results of seven language parsers, evaluated on the testsets of the UD corpus (v2.6) of the corresponding language. Parsers trained with the larger synthetic data showed higher LAS than those trained with the smaller manually created UD corpus data.

Figure 3 shows LAS against the size of training corpora. All languages show similar trends between parsing accuracy and corpus size; larger synthetic corpora compensate for the smaller size of the UD corpora, except for English, French and German in which the synthetic data performs nearly equally with the same size of the original UD training data.

4 Conclusion and Future Work

We presented a data augmentation approach for UD parsing that improves fast production parsers accuracy and overcomes critical treebank limitations. A new Transformer enhanced biaffine parser is used to generate scalable synthetic data. We showed that utilizing deep contextualized representations pretrained on massive multilingual corpora can be used to considerably improve parsing accuracy. In the future, we plan to extending our method to generate synthetic data for additional languages.

References

Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

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, November. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China, 22–24 Jun. PMLR.

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

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark, September. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Taesun Moon, Parul Awasthy, Jian Ni, and Radu Florian. 2019. Towards lingua franca named entity recognition with bert. *ArXiv*, abs/1912.01389.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May. European Language Resources Association.

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the workshop on incremental parsing: Bringing engineering and cognition together*, pages 50–57.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium, October. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Wilson L. Taylor. 1953. "cloze procedure": a new tool for measuring readability. *Journalism & Mass Communication Quarterly*, 30:415–433.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Daniel Zeman, Filip Ginter, Jan Hajič, Joakim Nivre, Martin Popel, and Milan Straka. 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*, Brussels, Belgium.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium, October-November. Association for Computational Linguistics.