Overcoming Early Saturation on Low-Resource Languages in Multilingual Dependency Parsing

Jiannan Mao[†*], **Chenchen Ding**[‡], **Hour Kaing**[‡],
Hideki Tanaka[‡], **Masao Utiyama**[‡], **Tadahiro Matsumoto**[†]
[†]Gifu University, Gifu, Japan
[‡]National Institute of Information and Communications Technology, Kyoto, Japan
[†]{mao, tad}@mat.info.gifu-u.ac.jp
[‡]{chenchen.ding, hour_kaing, hideki.tanaka, mutiyama}@nict.go.jp

Abstract

UDify (Kondratyuk and Straka, 2019) is a multilingual and multi-task parser fine-tuned on mBERT that achieves remarkable performance in high-resource languages. However, the performance saturates early and decreases gradually in low-resource languages as training proceeds. This work applies a data augmentation method and conducts experiments on seven few-shot and four zero-shot languages. The unlabeled attachment scores were improved on the zero-shot languages dependency parsing tasks, with the average score rising from 67.1% to 68.7%. Meanwhile, dependency parsing tasks for high-resource languages and other tasks were hardly affected. Experimental results indicate the data augmentation method is effective for low-resource languages in a multilingual dependency parsing.

Keywords: Parsing, Multilinguality, Low Resource Languages, Unsupervised Learning

1. Introduction

A dependency parser can be efficiently trained when large treebanks are available (Dozat and Manning, 2017; Qi et al., 2020). For low-resource languages with no (zero-shot) or limited (few-shot) treebanks, multilingual modeling has emerged as an efficient solution, where cross-lingual information is leveraged to compensate for the lack of data. Scholivet et al. (2019); Üstün et al. (2022) have demonstrated that the performance on multilingual tasks can be boosted by pairing languages with similarities. Multilingualism also reduces the expense when training multiple models for a group of languages (Johnson et al., 2017; Aharoni et al., 2019; Cai et al., 2021; Muennighoff et al., 2023).

UDify (Kondratyuk and Straka, 2019) is a multi-task network fine-tuned on multilingual BERT (mBERT) (Devlin et al., 2019) pre-trained embeddings. It is capable of producing annotations for any treebank from Universal Dependencies (UD) (Zeman et al., 2018). UDify exhibits strong and consistent performance across all 124 UD treebanks for 75 languages and multiple tasks such as lemmatization, part-of-speech (POS), and dependency parsing. However, an issue not yet paid enough attention in several related studies is the substantial discrepancy found in the performance of these methods in low-resource language learning scenarios, even when almost the identical training strategies, datasets, models, and evaluation methods were used in Choudhary (2021), Üstün et al. (2022), Effland and Collins (2023).

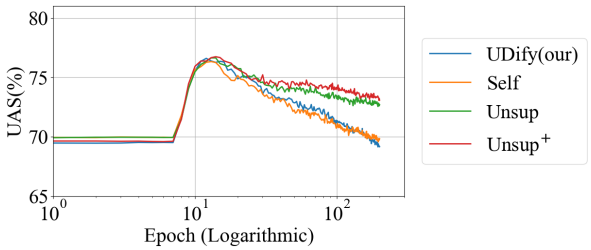

Figure 1: Change in the UAS(%) of a model during the training process on the Breton–KEB test set for both baselines: UDify(our) and Self, as well as the proposed method, Unsup and Unsup$^+$.

To address and investigate this issue, the work of Mao et al. (2023) conducts an experimental exploration into the low-resource case phenomenon by observing changes during model training. They adopted the data augmentation strategy, which leverages the original UDify for parsing raw sentences in single low-resource language to obtain initial probabilities. This is followed by the application of unsupervised learning to train these probabilities. Using the trained probabilities to create artificially structured dependency data and merging them into UDify's training set enables UDify to be trained on a more extensive dataset.

In this work, we conducted comprehensive experiments on low-resource languages using data augmentation methods, expanded (for few-shot languages) and created (for zero-shot languages) artificial treebanks for the seven few-shot and four

[*] This work was done during the first author's internship at National Institute of Information and Communications Technology, Kyoto, Japan.

Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD 2024)
@LREC-COLING-2024, pages 63–69. May 25, 2024.

zero-shot languages. By combining these artificial treebanks with the UD treebanks and using the UDify framework, we trained a multilingual parser. As a result, increases in the unlabeled attachment score (UAS) for zero-shot languages were observed, with the average value increasing from 67.1% to 68.7%; in the most-improved case, the UAS rocketed from 78.4% to 88.0%. Similarly, the few-shot languages experienced a UAS increase of 0.2%. In contrast, the UAS for other languages and evaluation scores for other tasks did not show significant changes, which suggests that the overall robustness of multilingual and multi-task processing is retained.

2. Background

2.1. UDify

The UDify model jointly predicts lemmas, POS tags, morphological features, and dependency structures. The pre-trained mBERT model[1] is used in the UDify model for cross-lingual learning without additional tags to distinguish the languages. In addition, a strategy similar to ELMo (Peters et al., 2018) is adopted, where a weighted sum of the outputs of all layers is computed as follows and fed to a task-specific classifier:

$$e_j^{task} = \sum_i mBERT_{ij}.$$

Here, e^{task} denotes the contextual output embeddings for tasks such as the dependency parse. In addition, $mBERT_{ij}$ denotes the $mBERT$ representation for layer i at token position j.

In the task involving dependency structures, mBERT's subword tokenization process inputs words into multiple subword units. However, only the embeddings e_j^{task} of the first subword unit are used, serving as input to the graph-based bi-affine attention classifier (Dozat and Manning, 2017). The resulting outputs are combined using bi-affine attention to produce a probability distribution of the arc-head for each word. Finally, the dependency tree is decoded using the Chu–Liu/Edmonds algorithm (Chu, 1965; Edmonds et al., 1967).

2.2. Unsupervised Dependency Learning

Adhering to the properties of dependency syntax (Robinson, 1970), a general unsupervised algorithm for projective N-gram dependency learning (Unsupervised-Dep) was described in Ding and Yamamoto (2013, 2014). This method constructs the best dependency tree with a dynamic programming method using a CYK style chart and is based on the complete-link and complete-sequence non-constituent concepts. However, considering the time complexity of this approach for arbitrary N-gram dependency learning, which may not be ideal for practical applications, we chose to focus in this study on the case of the bi-gram.

When considering the bi-gram, the directionality of a pair of words is set by the dependency relation, with ($w_i \rightarrow w_j$) indicating a rightward relation and ($w_i \leftarrow w_j$) indicating a leftward one. The bi-gram unsupervised learning update probabilities $P(w_i \rightarrow w_j)$ and $P(w_i \leftarrow w_j)$ are calculated using the Inside–Outside algorithm (Lari and Young, 1990). Finally, the Viterbi algorithm (Forney, 1973) is employed to determine the tree construction in the calculated Inside portion with the maximum probability, thus generating the optimal structure.

3. Investigation

3.1. UDify with Data Augmentation

In the work of Mao et al. (2023), a data augmentation based on Unsupervised-Dep is provided. Due to Unsupervised-Dep has a high time complexity of $O(n^3)$, making the common practice in the original methods, which start training from a random probability, somewhat inefficient. To circumvent this, the parsing results from UDify were utilized to initialize the probabilities. Despite the potential decrease in UDify's accuracy on low-resource languages during its training, the final results consistently outperform those from other parsing models (Qi et al., 2018; Tran and Bisazza, 2019), providing a solid foundation for the proposed initialization approach.

The process starts with the raw corpus, $Data$, input into the trained UDify by the original UD treebank, to generate the dependency arc-heads, represented as DEP_{arc}, and POS, lemmas, etc., denoted as $Others$. Statistical computations on DEP_{arc} generate initial probabilities $P(w_i \rightarrow w_j)$ and $P(w_i \leftarrow w_j)$, serving as input for Unsupervised-Dep alongside $Data$.

Following several iterations of training through Unsupervised-Dep, the re-estimated $P(w_i \rightarrow w_j)^{'}$ and $P(w_i \leftarrow w_j)^{'}$ emerge. They become the parameters for the Viterbi algorithm to determine the optimal dependency arc-head as given by

$$DEP_{arc}^{'} = Viterbi(x, P(w_i \rightarrow w_j)^{'}, P(w_i \leftarrow w_j)^{'}) \ ,$$

where $DEP_{arc}^{'}$ is the tree with the highest probability for a sentence x from $Data$.

Finally, $DEP_{arc}^{'}$ is merged with $Others$, ultimately generating artificial data. The artificial data are then combined with the existing UD treebanks for the subsequent UDify training.

3.2. On Few- and Zero-Shot Languages

During the training of UDify, the dependency structures for zero-shot languages are learned through

[1] github.com/google-research/bert/multilingual.md

transfer learning. Compared to high-resource languages, an early saturation in the accuracy of dependency parsing is observed across all zero-shot languages during the learning process. The peak performance is typically reached around the 12th training epoch, as illustrated in Figure 2. Mao et al. (2023) applied data augmentation to individual zero-shot language, effectively addressing this issue.

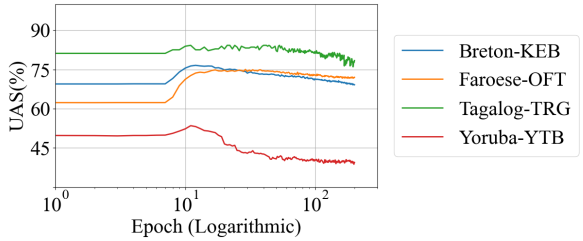

Figure 2: Change in the UAS(%) of low-resource languages during UDify(our) training.

However, when applying Unsupervised-Dep data augmentation to multiple zero-shot languages, the effectiveness of this approach has not been explored due to the impact of the amount of data generated on parser performance. Especially considering that this approach may generate large amounts of artificial data, its practical application in this context needs to be evaluated.

Moreover, the training of multilingual parser reveals that few-shot languages are similarly affected by the volume of training data. This highlights the critical need for effective data augmentation methods to improve the parsing performance of models like UDify. We aim to employ Unsupervised-Dep for multiple languages to explore its potential in mitigating early saturation in zero-shot languages and improving parsing accuracy in few-shot languages within a multilingual context.

4. Experiments

4.1. Dataset

The raw data of seven few-shot and four zero-shot languages that are most often tokenized using spaces were collected from El-Kishky et al. (2020); Fan et al. (2021); Schwenk et al. (2021) to create our selected low-resource language set for the implementation of Unsupervised-Dep. The data in the experiment are summarized in Table 1 and referred to as OPUS-mult in subsequent sections.

For comparison with the UDify and to illustrate our motivation, our parser experiments employed the UD Treebank v2.3 used by UDify. During training, following McDonald et al. (2011), we merged training sets, randomized the sentence order each epoch, and fed the network diverse batches of original and artificial data from multiple languages.

language(code)	#sent.(len.)	#train	#test
Armenian(hy)	2.4(8.2)	560	470
Belarusian(be)	2.0(9.0)	260	68
Hungarian(hu)	134.1(5.3)	910	449
Kazakh(kk)	1.7(8.2)	31	1,047
Lithuanian(lt)	236.7(5.6)	153	55
Marathi(mr)	1.5(10.0)	373	47
Tamil(ta)	13.7(7.7)	400	120
Breton(br)	18.2(9.5)	0	888
Faroese(fo)	1.3(8.1)	0	1,208
Tagalog(tl)	150.0(16.2)	0	55
Yoruba(yo)	9.7(8.1)	0	100

Table 1: Raw data collected from various corpora. Above: few-shot languages; below: zero-shot languages. #sent.(len.) denotes the raw sentences in unsupervised learning (in thousands), with the numbers in parentheses indicating the average length. #train and #test are the sentence counts in UD v2.3 treebank's training and test sets, respectively.

4.2. Setup

To minimize the impact of experimental environment variations on the result of Popel and Bojar (2018) in the comparisons, we followed the parameter settings from Kondratyuk and Straka (2019) and re-implemented the model as UDify(our). Additionally, to expedite the training process, we employed Horovod (Sergeev and Balso, 2018) to implement parallel computation.

At the beginning of training on Unsupervised-Dep, we used the UDify(our) model to parse each language present in the OPUS-mult dataset. The statistical results derived from the parsing outcomes of each language were adopted as its initial probabilities, which were continuously re-estimated throughout the unsupervised learning process. After the 10th training iteration, we employed the newly estimated probabilities to parse the sentences from OPUS-mult.

To assess the impact of augmenting training data for multiple low-resource languages on parsing accuracy, we designed and conducted several experiments. In the Unsupervised-Dep data augmentation experiments, we randomly selected 300 sentences for each language from OPUS-mult, processed them using Unsupervised-Dep, and integrated them into the UD treebanks to form the training dataset. The model trained from this dataset is referred to as Unsup. Inspired by the work of Rybak and Wróblewska (2018), we conducted a comparative experiment using a data augmentation method dubbed Self. In this approach, we used the same raw sentences train Unsup model and directly applied the parsing results obtained from the UDify(org) model. These results were merged with the original training set to train the Self model.

	hy	be	hu	kk	lt	mr	ta	br	fo	tl	yo	Few	Zero
UDify(org)	85.6	91.8	89.7	74.8	79.1	**79.4**	79.3	63.5	67.2	64.0	37.6	-	-
UDify(our)	86.1	92.1	89.8	76.0	79.4	74.3	80.8	69.2	72.0	78.4	39.4	84.0	67.1
Self	85.9	**92.5**	89.6	**76.2**	79.2	74.8	**81.2**	69.8	**72.5**	85.3	38.8	84.0	67.6
Unsup	**86.3**	92.4	**90.0**	**76.2**	**79.5**	74.0	80.5	**72.7**	71.9	**88.0**	**39.6**	**84.2**	**68.7**

Table 2: UAS(%) for few- and zero-shot languages obtained using different methods. The last two columns display the combined test set results for few- (Few) and for zero-shot (Zero) languages. We denote the treebank names using language codes; both the low-resource languages have only one treebank in UD v2.3. The UDify(org) result was reported in Kondratyuk and Straka (2019).

	Zero-shot		Other	
	UAS	Rest	UAS	Rest
UDify(our)	67.1	55.6	77.5	82.5
Self	67.6	56.3	77.5	82.4
Unsup	**68.7**	**59.0**	77.5	82.5

Table 3: UD scores on selected zero-shot and other languages obtained by different methods. Rest(%) refers to the average score of UPOS, UFeats, Lemma, and LAS in the UD scores.

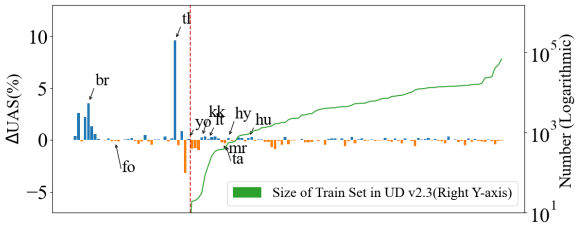

Figure 3: Difference in UAS(%) on all test treebanks: blue indicates Unsup $>$ UDify(our), orange indicates Unsup $<$ UDify(our). The left side of the red dotted line shows zero-shot languages.

4.3. Result and Discussion

A comparison with the experimental findings from Kondratyuk and Straka (2019) confirms the successful re-implementation of UDify(our), as illustrated in Table 2, and reveals that our replicated model surpasses those in related work (Choudhary, 2021; Üstün et al., 2022; Mao et al., 2023). Although no method produced a noticeable improvement for the few-shot languages, the results in this table indicate a significant improvement in UDify's ability to parse the dependency arc-head accuracy for zero-shot languages at the end of the training with the Unsupervised-Dep data augmentation method. This is reflected in the results for the combined test set, where the UAS increased to 68.7%. Taking Breton from the zero-shot languages as an example, we illustrate the changes in UAS during the training process under different methods in Figure 1. The figure reveals that the inclusion of data generated through Unsupervised-Dep significantly mitigates the reduction in UAS accuracy for zero-shot languages over the course of the training, thereby improving the result.

The UAS of almost every zero-shot language improved when artificial data via Unsupervised-Dep were included. To our knowledge, this is the state-of-the-art result for Tagalog. The `Tagalog-TRG` treebank is quite small, encompassing only 55 sentences with an average sentence length of 4.2 words in UD v2.3. In contrast, we have gathered 150k Tagalog raw sentences with an average length of 16.2 words. We believe that the quality and quantity of raw sentences used for training Unsupervised-Dep have a crucial impact on the performance of the multilingual parser.

To further enhance UDify's dependency parsing accuracy in low-resource languages, we attempted to increase the number of sentences generated by Unsupervised-Dep data augmentatio to 500, which we refer to as Unsup$^+$. In the result of Unsup$^+$, the UAS of the selected zero-shot languages in the test set saw further improvement, reaching 69.3%. We depict the changes in UAS for Breton during the Unsup$^+$ training process in Figure 1.

Given UDify's standing as a multilingual and multi-task parser, assessing the impact of our proposed methods on other languages and tasks is essential. To further scrutinize the variations between the UAS results of UDify(org) and Unsup, we carried out tests on all treebanks. As shown in Figure 3, the results indicate that Unsup effectively enhanced the UAS of zero-shot languages when artificial data were created using Unsupervised-Dep, especially for Breton and Tagalog. Meanwhile, its impact on the parsing precision of dependency structures in other languages is negligible.

For a comprehensive comparison, the UD scores of the zero-shot and other languages have been compiled in Table 3. Given that UDify must balance the loss produced by multiple decoders during training and the work of Rybak and Wróblewska (2018), these variations in evaluation metrics are considered reasonable. Broadly, our method has not had a negative impact on other languages and tasks, maintaining their performance levels.

Considering all results, we argue that creating training data for multiple low-resource languages using Unsupervised-Dep is both essential and effective in multilingual modeling contexts.

5. Conclusion and Future Work

This study highlights the issue of early saturation in parsing accuracy for UDify across multiple low-resource languages. To address this challenge, we implemented data augmentation for several low-resource languages through unsupervised learning. The experimental results demonstrated the effectiveness of data augmentation method in enhancing the parsing performance of multilingual parsers for low-resource languages.

Despite the limitations posed by training speed and the quality and quantity of raw data on our experiments, two possibilities remain: (1) Generating more data for zero-shot languages could lead to positive improvements. (2) The quality and quantity of raw data play a crucial role in the effectiveness of unsupervised data augmentation methods, thereby affecting the performance of multilingual parsers.

In future work, our research aims to explore additional influencing factors and considerations to further enhance multilingual parsing performance in low-resource language scenarios. Moreover, we plan to conduct research and exploration on low-resource languages using the latest UD treebanks.

6. Bibliographical References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. 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 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Deng Cai, Xin Li, Jackie Chun-Sing Ho, Lidong Bing, and Wai Lam. 2021. Multilingual AMR parsing with noisy knowledge distillation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2778–2789, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chinmay Choudhary. 2021. Improving the performance of UDify with linguistic typology knowledge. In *Proceedings of the Third Workshop on Computational Typology and Multilingual NLP*, pages 38–60, Online. Association for Computational Linguistics.

Yoeng-Jin Chu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

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.

Chenchen Ding and Mikio Yamamoto. 2013. An unsupervised parameter estimation algorithm for a generative dependency n-gram language model. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 516–524, Nagoya, Japan. Asian Federation of Natural Language Processing.

Chenchen Ding and Mikio Yamamoto. 2014. A generative dependency n-gram language model: Unsupervised parameter estimation and application. *Information and Media Technologies*, 9(4):857–885.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jack Edmonds et al. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Thomas Effland and Michael Collins. 2023. Improving low-resource cross-lingual parsing with expected statistic regularization. *Transactions of the Association for Computational Linguistics*, 11:122–138.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1).

G David Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

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.

Karim Lari and Steve J Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 4(1):35–56.

Jiannan Mao, Chenchen Ding, Hour Kaing, Hideki Tanaka, Masao Utiyama, and Tadahiro Matsumoto. 2023. Improving zero-shot dependency parsing by unsupervised learning. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 217–226, Hong Kong, China. Association for Computational Linguistics.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Matthew E. 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. Association for Computational Linguistics.

Martin Popel and Ondrej Bojar. 2018. Training tips for the transformer model. *Prague Bull. Math. Linguistics*, 110:43–70.

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. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Jane J. Robinson. 1970. Dependency structures and transformational rules. *Language*, 46(2):259–285.

Piotr Rybak and Alina Wróblewska. 2018. Semi-supervised neural system for tagging, parsing and lematization. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 45–54, Brussels, Belgium. Association for Computational Linguistics.

Manon Scholivet, Franck Dary, Alexis Nasr, Benoit Favre, and Carlos Ramisch. 2019. Typological features for multilingual delexicalised 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 3919–3930, Minneapolis, Minnesota. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Alexander Sergeev and Mike Del Balso. 2018. Horovod: Fast and easy distributed deep

learning in TensorFlow. *arXiv preprint arXiv:1802.05799*.

Ke Tran and Arianna Bisazza. 2019. Zero-shot dependency parsing with pre-trained multilingual sentence representations. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 281–288, Hong Kong, China. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2022. UDapter: Typology-based language adapters for multilingual dependency parsing and sequence labeling. *Computational Linguistics*, 48(3):555–592.

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.