Multilingual Constituency Parsing with Self-Attention and Pre-Training

Nikita Kitaev **Steven Cao** **Dan Klein**
Computer Science Division
University of California, Berkeley
{kitaev,stevencao,klein}@berkeley.edu

Abstract

We show that constituency parsing benefits from unsupervised pre-training across a variety of languages and a range of pre-training conditions. We first compare the benefits of no pre-training, fastText (Bojanowski et al., 2017; Mikolov et al., 2018), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2018a) for English and find that BERT outperforms ELMo, in large part due to increased model capacity, whereas ELMo in turn outperforms the non-contextual fastText embeddings. We also find that pre-training is beneficial across all 11 languages tested; however, large model sizes (more than 100 million parameters) make it computationally expensive to train separate models for each language. To address this shortcoming, we show that joint multilingual pre-training and fine-tuning allows sharing all but a small number of parameters between ten languages in the final model. The 10x reduction in model size compared to fine-tuning one model per language causes only a 3.2% relative error increase in aggregate. We further explore the idea of joint fine-tuning and show that it gives low-resource languages a way to benefit from the larger datasets of other languages. Finally, we demonstrate new state-of-the-art results for 11 languages, including English (95.8 F1) and Chinese (91.8 F1).

1 Introduction

There has recently been rapid progress in developing contextual word representations that improve accuracy across a range of natural language tasks (Peters et al., 2018; Howard and Ruder, 2018; Radford et al., 2018; Devlin et al., 2018a). While we have shown in previous work (Kitaev and Klein, 2018) that such representations are beneficial for constituency parsing, our earlier results only consider the LSTM-based ELMo representations (Peters et al., 2018), and only for the English language. In this work, we study a broader range of pre-training conditions and experiment over a variety of languages, both jointly and individually.

First, we consider the impact on parsing of using different methods for pre-training initial network layers on a large collection of un-annotated text. Here, we see that pre-training provides benefits for all languages evaluated, and that BERT (Devlin et al., 2018a) outperforms ELMo, which in turn outperforms fastText (Bojanowski et al., 2017; Mikolov et al., 2018), which performs slightly better than the non pre-trained baselines. Pre-training with a larger model capacity typically leads to higher parsing accuracies.

Second, we consider various schemes for the parser fine-tuning that is required after pre-training. While BERT itself can be pre-trained jointly on many languages, successfully applying it, e.g. to parsing, requires task-specific adaptation via fine-tuning (Devlin et al., 2018a). Therefore, the obvious approach to parsing ten languages is to fine-tune ten times, producing ten variants of the parameter-heavy BERT layers. In this work, we compare this naive independent approach to a joint fine-tuning method where a single copy of fine-tuned BERT parameters is shared across all ten languages. Since only a small output-specific fragment of the network is unique to each task, the model is 10x smaller while losing an average of only 0.28 F1.

Although, in general, jointly training multilingual parsers mostly provides a more compact model, it does in some cases improve accuracy as well. To investigate when joint training is helpful, we also perform paired fine-tuning on all pairs of languages and examine which pairs lead to the largest increase in accuracy. We find that larger treebanks function better as auxiliary tasks and that only smaller treebanks see a benefit from joint training. These results suggest that this manner of joint training can be used to provide support for many languages in a resource-efficient man-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3499–3505
Florence, Italy, July 28 - August 2, 2019.

ner, but does not exhibit substantial cross-lingual generalization except when labeled data is limited. Our parser code and trained models for eleven languages are publicly available.[1]

2 Model

Our parsing model is based on the architecture described in Kitaev and Klein (2018), which is state of the art for multiple languages, including English. A constituency tree T is represented as a set of labeled spans,

$$T = \{(i_t, j_t, l_t) : t = 1, \ldots, |T|\}$$

where the t^{th} span begins at position i_t, ends at position j_t, and has label l_t. The parser assigns a score $s(T)$ to each tree, which decomposes as

$$s(T) = \sum_{(i,j,l) \in T} s(i, j, l)$$

The per-span scores $s(i, j, l)$ are produced by a neural network. This network accepts as input a sequence of vectors corresponding to words in a sentence and transforms these representations using one or more self-attention layers. For each span (i, j) in the sentence, a hidden vector $v_{i,j}$ is constructed by subtracting the representations associated with the start and end of the span. An MLP span classifier, consisting of two fully-connected layers with one ReLU nonlinearity, assigns labeling scores $s(i, j, \cdot)$ to the span. Finally, the the highest scoring valid tree

$$\hat{T} = \arg\max_{T} s(T)$$

can be found efficiently using a variant of the CKY algorithm. For more details, see Kitaev and Klein (2018).

We incorporate BERT by computing token representations from the last layer of a BERT model, applying a learned projection matrix, and then passing them as input to the parser. BERT associates vectors to sub-word units based on WordPiece tokenization (Wu et al., 2016), from which we extract word-aligned representations by only retaining the BERT vectors corresponding to the last sub-word unit for each word in the sentence. We briefly experimented with other alternatives, such as using only the first sub-word instead, but did not find that this choice had a substantial effect on English parsing accuracy.

[1] https://github.com/nikitakit/self-attentive-parser

Method	Pre-trained on	Params	F1
No pre-training	–	26M	93.61[a]
FastText	English	626M	93.72
ELMo	English	107M	95.21[a]
$\text{BERT}_{\text{BASE}}$ (uncased)	Chinese	110M	93.57
$\text{BERT}_{\text{BASE}}$ (cased)	104 languages	185M	94.97
$\text{BERT}_{\text{BASE}}$ (uncased)	English	117M	95.32
$\text{BERT}_{\text{BASE}}$ (cased)	English	116M	95.24
$\text{BERT}_{\text{LARGE}}$ (uncased)	English	343M	95.66
$\text{BERT}_{\text{LARGE}}$ (cased)	English	341M	95.70
Ensemble (final 4 models above)		916M	**95.87**

Table 1: Comparison of parsing accuracy on the WSJ development set when using different word representations. [a]Kitaev and Klein (2018)

The fact that additional layers are applied to the output of BERT – which itself uses a self-attentive architecture – may at first seem redundant, but there are important differences between these two portions of the architecture. The extra layers on top of BERT use word-based tokenization instead of sub-words, apply the factored version of self-attention proposed in Kitaev and Klein (2018), and are randomly-initialized instead of being pre-trained. We found that passing the (projected) BERT vectors directly to the MLP span classifier hurts parsing accuracies.

We train our parser with a learning rate of 5×10^{-5} and batch size 32, where BERT parameters are fine-tuned as part of training. We use two additional self-attention layers following BERT. All other hyperparameters are unchanged from Kitaev and Klein (2018) and Devlin et al. (2018a).

3 Comparison of Pre-Training Methods

In this section, we compare using BERT, ELMo, fastText, and training a parser from scratch on treebank data alone. Our comparison of the different methods for English is shown in Table 1. $\text{BERT}_{\text{BASE}}$ ($\sim$115M parameters) performs comparably or slightly better than ELMo ($\sim$107M parameters; 95.32 vs. 95.21 F1), while $\text{BERT}_{\text{LARGE}}$ ($\sim$340M parameters) leads to better parsing accuracy (95.70 F1). Furthermore, both pre-trained contextual embeddings significantly outperform fastText, which performs slightly better than no pre-training (93.72 vs. 93.61 F1). These results show that both the LSTM-based architecture of ELMo and the self-attentive architecture of BERT are viable for parsing, and that pre-training benefits from having a high model capacity. We did not

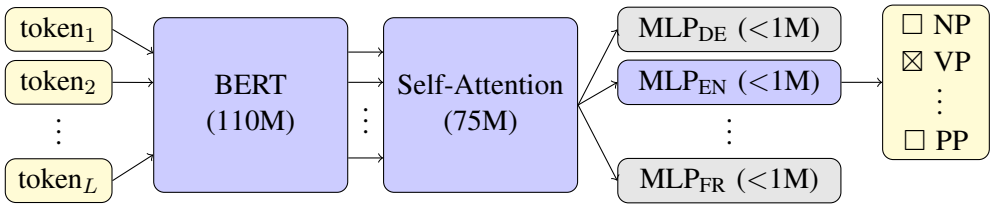

Figure 1: The architecture of the multilingual model, with components labeled by the number of parameters.

observe a sizable difference between an "uncased" version of BERT that converts all text to lowercase and a "cased" version of that retains case information.

We also evaluate an ensemble of four English BERT-based parsers, where the models are combined by averaging their span label scores:

$$s_{\text{ensemble}}(i,j,l) = \frac{1}{4}\sum_{n=1}^{4} s_n(i,j,l)$$

The resulting accuracy increase with respect to the best single model (95.87 F1 vs. 95.66 F1) reflects not only randomness during fine-tuning, but also variations between different versions of BERT. When combined with the observation that $\text{BERT}_{\text{LARGE}}$ outperforms $\text{BERT}_{\text{BASE}}$, the ensemble results suggest that empirical gains from pre-training have not yet plateaued as a function of computational resources and model size.

Next, we compare pre-training on monolingual data to pre-training on data that includes a variety of languages. We find that pre-training on only English outperforms multilingual pre-training given the same model capacity, but the decrease in accuracy is less than 0.3 F1 (95.24 vs. 94.97 F1). This is a promising result because it supports the idea of parameter sharing as a way to provide support for many languages in a resource-efficient manner, which we examine further in Section 4.

To further examine the effects of pre-training on disparate languages, we consider the extreme case of training an English parser using a version of BERT that was pre-trained on the Chinese Wikipedia. Neither the pre-training data nor the subword vocabulary used are a good fit for the target task. However, English words (e.g. proper names) occur in the Chinese Wikipedia data with sufficient frequency that the model can losslessly represent English text: all English letters are included in its subword vocabulary, so in the worst case it will decompose an English word into its individual letters. We found that this model achieves performance comparable to our earlier parser (Kitaev and Klein, 2018) trained on treebank data alone (93.57 vs. 93.61 F1). These results suggest that even when the pre-training data is a highly imperfect fit for the target application, fine-tuning can still produce results better than or comparable to purely supervised training with randomly-initialized parameters.[2]

4 Multilingual Model

We next evaluate how well self-attention and pre-training work cross-linguistically; for this purpose we consider ten languages: English and the nine languages represented in the SPMRL 2013/2014 shared tasks (Seddah et al., 2013).

Our findings from the previous section show that pre-training continues to benefit from larger model sizes when data is abundant. However, as models grow, it is not scalable to conduct separate pre-training and fine-tuning for all languages. This shortcoming can be partially overcome by pre-training BERT on multiple languages, as suggested by the effectiveness of the English parser fine-tuned from multilingual BERT (see Table 1). Nevertheless, this straightforward approach also faces scalability challenges because it requires training an independent parser for each language, which results in over 1.8 billion parameters for ten languages. Therefore, we consider a single parser with parameters shared across languages and fine-tuned jointly. The joint parser uses the same BERT model and self-attention layers for all ten languages but contains one MLP span classifier per language to accommodate the different tree labels (see Figure 1). The MLP layers contain 250K-850K parameters, depending on the type of syntactic annotation adopted for the language, which

[2]We also attempted to use a randomly-initialized BERT model, but the resulting parser did not train effectively within the range of hyperparameters we tried. Note that the original BERT models were trained on significantly more powerful hardware and for a longer period of time than any of the experiments we report in this paper.

	Arabic	Basque	English	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	Avg	Params
No pre-training[a]	85.61	89.71	93.55	84.06	87.69	90.35	92.69	86.59	93.69	84.45	88.32	355M
One model per language (this work)	**87.97**	**91.63**	**94.91**	**87.42**	**90.20**	**92.99**	**94.90**	88.80	**96.36**	88.86	**91.40**	1,851M
Joint multilingual model (this work)	87.44	90.70	94.63	87.35	88.40	92.95	94.60	**88.96**	96.26	**89.94**	91.12	189M
Relative ΔError vs. monolingual	+4.2%*	+10.0%*	+5.2%*	+0.6%	+15.5%*	+0.6%	+5.6%*	-1.5%	+2.7%	-10.7%*	+3.2%*	

Table 2: Results of monolingual and multilingual training on the SPMRL and WSJ test splits using the version of BERT pre-trained on 104 languages. In the last row, starred differences are significant at the $p < 0.05$ level using a bootstrap test; see Berg-Kirkpatrick et al. (2012). [a]Kitaev and Klein (2018)

Auxiliary Language	Arabic	Basque	English	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	Average	Best	Best Aux.
# train sentences	15,762	7,577	39,831	14,759	40,472	5,000	8,146	23,010	6,578	5,000			
Language Tested													
Arabic	0	-0.38	-0.20	-0.27	-0.26	-0.14	-0.29	-0.13	-0.31	-0.33	-0.23	**+0**	None
Basque	-0.47	0	-0.06	-0.26	0.04	-0.22	-0.27	-0.41	-0.49	-0.34	-0.25	**+0.04**	German
English	-0.18	-0.04	0	-0.02	-0.03	-0.07	-0.09	0.05	0.10	-0.05	-0.03	**+0.10**	Polish
French	0.42	0.01	0.28	0	0.40	-0.14	0.04	0.27	0.29	-0.10	0.15	**+0.42***	Arabic
German	-0.38	-0.20	0.03	-0.45	0	-0.13	-0.15	-0.13	-0.21	-0.26	-0.19	**+0.03**	English
Hebrew	0.13	0.05	-0.27	-0.17	-0.11	0	-0.09	-0.19	-0.30	-0.35	-0.13	**+0.13**	Arabic
Hungarian	-0.14	-0.43	-0.29	-0.38	-0.11	-0.39	0	-0.17	-0.28	-0.32	-0.25	**+0**	None
Korean	-0.24	-0.25	0.16	-0.27	-0.11	-0.01	0	0	-0.07	-0.17	-0.10	**+0.16**	English
Polish	0.25	0.15	0.20	0.24	0.24	0.21	0.14	0.20	0	0.12	0.18	**+0.25***	Arabic
Swedish	0.17	-0.08	0.38	0.54	0.53	-0.11	0.59	0.78	-0.17	0	0.26	**+0.78***	Korean
Average	-0.04	-0.12	0.02	-0.10	0.06	-0.10	-0.01	0.03	-0.14	-0.18			

Table 3: Change in development set F1 score due to paired vs. individual fine-tuning. In the "Best" column, starred results are significant at the $p < 0.05$ level. On average, the three largest treebanks (German, English, Korean) function the best as auxiliaries. Also, the three languages benefitting most from paired training (Swedish, French, Polish) function poorly as auxiliaries.

is less than 0.5% of the total parameters. Therefore, this joint training entails a 10x reduction in model size.

During joint fine-tuning, each batch contains sentences from every language. Each sentence passes through the shared layers and then through the MLP span classifier corresponding to its language. To reduce over-representation of languages with large training sets, we follow Devlin et al. (2018b) and determine the sampling proportions through exponential smoothing: if a language is some fraction f of the joint training set, the probability of sampling examples from that language is proportional to f^a for some a. We use the same hyperparameters as in monolingual training but increase the batch size to 256 to account for the increase in the number of languages, and we use $a = 0.7$ as in Devlin et al. (2018b). The individually fine-tuned parsers also use the same hyperparameters, but without the increase in batch size.

Table 2 presents a comparison of different parsing approaches across a set of ten languages. Our joint multilingual model outperforms treebank-only models (Kitaev and Klein, 2018) for each of the languages (88.32 vs 91.12 average F1). We also compare joint and individual fine-tuning. The multilingual model on average degrades performance only slightly (91.12 vs. 91.40 F1) despite the sharp model size reduction, and in fact performs better for Swedish.

We hypothesize that the gains/losses in accuracy for different languages stem from two competing effects: the multilingual model has access to more data, but there are now multiple objective functions competing over the same parameters. To examine language compatibility, we also train a bilingual model for each language pair and compare it to the corresponding monolingual model (see Table 3). From this experiment, we see that the best language pairs often do not correspond to any known linguistic groupings, suggesting that compatibility of objective functions is influenced more by other factors such as treebank labeling convention. In addition, we see that on average, the three languages with the largest training sets (English, German, Korean) function well as auxiliaries. Furthermore, the three languages that gain the most from paired training (Swedish, French, Polish) have smaller datasets and function poorly as auxiliaries. These results suggest that joint training not only drastically reduces model size, but also gives languages with small datasets a way to benefit from the large datasets of other languages.

	Arabic	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	Avg
Björkelund et al. (2014)	81.32[a]	88.24	82.53	81.66	89.80	91.72	83.81	90.50	85.50	86.12
Coavoux and Crabbé (2017)	82.92[b]	88.81	82.49	85.34	89.87	92.34	86.04	93.64	84.0	87.27
Kitaev and Klein (2018)	85.61[c]	89.71[c]	84.06	87.69	90.35	92.69	86.59[c]	93.69[c]	84.45	88.32
This work (joint multilingual model)	87.44	90.70	87.35	88.40	92.95	94.60	**88.96**	96.26	**89.94**	90.73
Δ vs. best previous	+1.83	+0.99	+3.29	+0.71	+2.60	+1.91	**+2.37**	+2.57	**+4.44**	
This work (one model per language)	**87.97**	**91.63**	**87.42**	**90.20**	**92.99**	**94.90**	88.80	**96.36**	88.86	**91.01**
Δ vs. best previous	**+2.36**	**+1.92**	**+3.36**	**+2.51**	**+2.64**	**+2.21**	+2.21	**+2.67**	+3.36	

Table 4: Results on the testing splits of the SPMRL dataset. All values are F1 scores calculated using the version of `evalb` distributed with the shared task. [a]Björkelund et al. (2013) [b]Uses character LSTM, whereas other results from Coavoux and Crabbé (2017) use predicted part-of-speech tags. [c]Does not use word embeddings, unlike other results from Kitaev and Klein (2018).

	LR	LP	F1
Dyer et al. (2016)	–	–	93.3
Choe and Charniak (2016)	–	–	93.8
Liu and Zhang (2017)	–	–	94.2
Fried et al. (2017)	–	–	94.66
Joshi et al. (2018)	93.8	94.8	94.3
Kitaev and Klein (2018)	94.85	95.40	95.13
This work (single model)	95.46	95.73	95.59
This work (ensemble of 4)	**95.51**	**96.03**	**95.77**

Table 5: Comparison of F1 scores on the WSJ test set.

	LR	LP	F1
Fried and Klein (2018)	–	–	87.0
Teng and Zhang (2018)	87.1	87.5	87.3
This work	**91.55**	**91.96**	**91.75**

Table 6: Comparison of F1 scores on the Chinese Treebank 5.1 test set.

5 Results

We train and evaluate our parsers on treebanks for eleven languages: the nine languages represented in the SPMRL 2013/2014 shared tasks (Seddah et al., 2013) (see Table 4), English (see Table 5), and Chinese (see Table 6). The English and Chinese parsers use fully monolingual training, while the remaining parsers incorporate a version of BERT pre-trained jointly on 104 languages. For each of these languages, we obtain a higher F1 score than any past systems we are aware of.

In the case of SPRML, both our single multilingual model and our individual monolingual models achieve higher parsing accuracies than previous systems (none of which made use of pre-trained contextual word representations). This result shows that pre-training is beneficial even when model parameters are shared heavily across languages.

6 Conclusion

The remarkable effectiveness of unsupervised pre-training of vector representations of language suggests that future advances in this area can continue improving the ability of machine learning methods to model syntax (as well as other aspects of language). As pre-trained models become increasingly higher-capacity, joint multilingual training is a promising approach to scalably providing NLP systems for a large set of languages.

Acknowledgments

This research was supported by DARPA through the XAI program. This work used the Savio computational cluster provided by the Berkeley Research Computing program at the University of California, Berkeley.

References

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics.

Anders Björkelund, Ozlem Cetinoglu, Agnieszka Faleńska, Richárd Farkas, Thomas Mueller, Wolfgang Seeker, and Zsolt Szántó. 2014. The IMS-Wrocław-Szeged-CIS entry at the SPMRL 2014 shared task: Reranking and morphosyntax meet unlabeled data. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 97–102.

Anders Björkelund, Ozlem Cetinoglu, Richárd Farkas, Thomas Mueller, and Wolfgang Seeker. 2013. (Re)ranking meets morphosyntax: State-of-the-art results from the SPMRL 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 135–145, Seattle, Washington, USA. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Do Kook Choe and Eugene Charniak. 2016. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336. Association for Computational Linguistics.

Maximin Coavoux and Benoit Crabbé. 2017. Multilingual lexicalized constituency parsing with word-level auxiliary tasks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 331–336. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805 [cs]*. ArXiv: 1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. BERT: Pre-training of deep bidirectional transformers for language understanding. https://github.com/google-research/bert/blob/master/multilingual.md.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209. Association for Computational Linguistics.

Daniel Fried and Dan Klein. 2018. Policy gradient as a proxy for dynamic oracles in constituency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 469–476. Association for Computational Linguistics.

Daniel Fried, Mitchell Stern, and Dan Klein. 2017. Improving neural parsing by disentangling model combination and reranking effects. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–166. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339. Association for Computational Linguistics.

Vidur Joshi, Matthew Peters, and Mark Hopkins. 2018. Extending a parser to distant domains using a few dozen partially annotated examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1199. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686. Association for Computational Linguistics.

Jiangming Liu and Yue Zhang. 2017. In-order transition-based constituent parsing. *Transactions of the Association for Computational Linguistics*, 5:413–424.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

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. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182. Association for Computational Linguistics.

Zhiyang Teng and Yue Zhang. 2018. Two local models for neural constituent parsing. In *Proceedings*

of the 27th International Conference on Computational Linguistics, pages 119–132. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv:1609.08144 [cs]*. ArXiv: 1609.08144.