Unsupervised and Few-Shot Parsing from Pretrained Language Models (Extended Abstract)[*]

Zhiyuan Zeng and **Deyi Xiong**
Tianjin University
{zhiyuan_zeng, dyxiong}@tju.edu.cn

Abstract

This paper proposes two Unsupervised constituent Parsing models (UPOA and UPIO) that calculate inside and outside association scores solely based on the self-attention weight matrix learned in a pretrained language model. The proposed unsupervised parsing models are further extended to few-shot parsing models (FPOA, FPIO) that use a few annotated trees to fine-tune the linear projection matrices in self-attention. Experiments on PTB and SPRML show that both unsupervised and few-shot parsing methods are better than or comparable to the previous methods.

1 Introduction

Automatically parsing a sentence to unveil the latent syntactic structure of the sentence is a long-standing task in natural language processing. Supervised syntactic parsing usually requires manually-annotated syntactic trees from a large treebank for training. However, building a treebank like PTB [Dahlmeier *et al.*, 2013] is expensive and time-consuming. Therefore, Unsupervised constituent parsing, which learns underlying structures without using any annotated trees, becomes an alternative to supervised syntactic structure learning [Shen *et al.*, 2018a; Shen *et al.*, 2019; Drozdov *et al.*, 2019; Kim *et al.*, 2019b; Kim *et al.*, 2019a; Wang *et al.*, 2019].

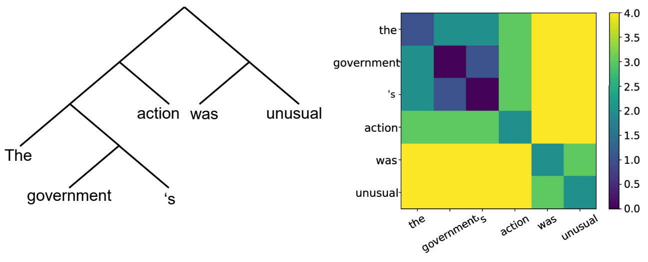
In this paper, we propose a new framework for unsupervised constituent parsing. First, we define a new syntactic distance for Unsupervised constituent Parsing, which is calculated according to an Outside Association score solely based on self-attention weight matrix: **UPOA**. Previous findings from many probing studies suggest that pretrained language models [Devlin *et al.*, 2019; Liu *et al.*, 2019; Radford *et al.*, 2019] are able to learn and embed latent syntactic structures in their parameters in an implicit way [Tenney *et al.*, 2019; Jawahar *et al.*, 2019]. Therefore, in UPOA, we exploit the self-attention weight matrix in BERT [Devlin *et al.*, 2019] to uncover latent syntactic structures of sentences. Particularly, we estimate the syntactic distance between two adjacent words as the negative self-attention weight between two adjacent spans that consume the two words. With the estimated syntactic distance, UPOA splits a span at the split point with the largest syntactic distance. However the syntactic distance defined in UPOA only considers the association between adjacent spans (outside association), ignoring the association among words inside a span (inside association). Therefore, we further propose an enhanced Unsupervised Parsing model **UPIO**, which splits a span according to the strength of both Inside and Outside association. The inside association is also estimated with the self-attention weights. With the estimated inside association and outside association, we can build a syntactic tree for a sentence with a greedy or chart-based parsing algorithm.

Second, we extend the UPIO to **FPIO**, a few-shot version of UPIO that can learn substantially better syntactic structures to narrow the performance gap between unsupervised and supervised parsing with just a few annotated trees. The fundamental idea behind FPIO is based on our finding with UPIO that the two linear projection matrices used by the query and key in the self-attention mechanism have a great impact on the parsing accuracy of the FPIO. We therefore freeze other parameters in BERT and propose a method to retrain (i.e., fine-tune) the two projection matrices on a few annotated trees. Similarly, we extend the UPOA to FPOA, a few-shot version of UPOA.

We carry the unsupervised and few-shot parsing experiments on the Penn Treebank (PTB) [Dahlmeier *et al.*, 2013] and SPMRL [Seddah and others, 2013] dataset. Our contributions can be summarized as follows:

- We propose an unsupervised constituent parsing model UPOA that calculates an out association score for span segmentation solely based on the self-attention weight matrix in a pretrained language model, and further propose an enhanced model, UPIO, which exploits both inside and outside association scores for estimating the likelihood of a span.
- We further extend our unsupervised parsing models UPOA and UPIO to few-shot learning methods FPOA and FPIO, which, trained on just a few annotated examples, substantially outperforms the few-shot and supervised parsing methods on the Penn Treebank and most languages of SPMRL.

[*] The full paper is published in Artificial Intelligence 305 (2022) 103665.



(a) (b)

Figure 1: The constituent tree (a) and the distance matrix corresponding to it (b) of an example sentence “The government ’s action was unusual”.

- We conduct experiments and in-depth analyses on the proposed unsupervised and few-shot parsing models, which not only demonstrate the effectiveness of both models, but also provide interesting findings, e.g., those with few-shot parsing on different layers and self-attention heads.

2 Unsupervised Parsing From Pretrained Self-Attention

In this section, we introduce two unsupervised parsing methods (UPOA and UPIO) to compute split scores based on the self-attention weights from a pretrained language model.

2.1 Estimating Split Scores Based on Syntactic Distance

We first introduce the syntactic distance presented by Shen *et al.* [2018a] and propose a new approach to calculating syntactic distance for unsupervised parsing from a distance matrix.

According to Shen *et al.* [2018a], the syntactic distance of two leaves is defined as the height of the lowest common ancestors of these two leaves. For the unlabeled constituent tree in Figure 1a, the syntactic distance of leaf “The” and leaf “government” is 2, while the syntactic distance of leaf “government” and leaf “’s” is 1. Shen *et al.* [2018b] propose a top-down parsing method that splits a span according to the syntactic distance between two adjacent leaves. A span is split between two adjacent words with the largest syntactic distance. The large syntactic distance indicates the two leaves share few common ancestors.

The syntactic distance defined in the aforementioned way shares some common properties with the dot product of two vectors. Firstly, syntactic distance measures the dissimilarity between two words according to the number of their common ancestors in the tree, while the dot product can be used to measure the similarity between two words with their vector representations. Secondly, a leaf node has the smallest syntactic distance with itself, while a vector has a high dot product with itself. Therefore we use the dot product of vectors to approximate the negative syntactic distance[1] of two words. The self-attention matrix of every head of a pretrained language model (particularly BERT used in this paper) contains normalized dot products of different word representations. Therefore every attention matrix can be taken as an approximation to the negative syntactic distance matrix.

If two spans have no intersection, the syntactic distance from any word in one span to any word in the other span is the same. For example in Figure 1b, the syntactic distances from any word in the span “The government ’s action” to any word in the span “was unusual” are all 4. Although we can take the negative attention weight of two adjacent words as the syntactic distance and apply the top-down algorithm proposed by Shen *et al.* [2018b] for unsupervised parsing, it is better to exploit the attention weights between two adjacent spans, instead of adjacent words, to estimate the syntactic distance, which can reduce the estimation bias of of syntactic distance. Given two adjacent spans: span(x, y), span$(y + 1, z)$, we average the negative attention weights from words in one span to words in the other span as the syntactic distance between these two spans:

$$d(\text{span}_{(x,y)}, \text{span}_{(y+1,z)}) = -\frac{\sum\limits_{i=x}^{y}\sum\limits_{j=y+1}^{z} a_{ij} + \sum\limits_{i=y+1}^{z}\sum\limits_{j=x}^{y} a_{ij}}{2(y-x+1)(z-y)} \tag{1}$$

where a_{ij} is the attention weight that word i attends to word j. We can take the syntactic distance between span(x, y) and span$(y + 1, z)$ as the split score of split point y in span(x, z), and parse a constituent tree from a self-attention weight matrix with a greedy or chart-based parsing algorithm.

2.2 Estimating Split Scores Based on the Strength of Inside and Outside Association

In a constituent tree, intuitively the strength of the association among words inside a constituent (**inside association**) is stronger than that of association to words outside the constituent (**outside association**). The syntactic distance can be considered to be related with the outside association, which measures the distance between two adjacent spans. However, it does not consider the inside association, which measures the distance between words inside the span.

Based on the aforementioned intuition, we compute the split score according to the inside association together with outside association. We first define the span score which measures how likely a span can be a constituent. For this, we again resort to the self-attention weights of pretrained language models, as they can be regarded as a relatedness metric between words. Given a span (x, y) from position x to y, the stronger the inside association and the weaker the outside association, the more possible that span (x, y) is functioning as a constituent. Therefore we estimate two scores for (x, y): $s_{\text{in}}(x, y)$ measuring the strength of the inside association and $s_{\text{out}}(x, y)$ for the outside association. We define $s_{\text{in}}(x, y)$ as the average attention weights a_{ij} inside the span which can be formulated as follows:

$$s_{\text{in}}(x, y) = \frac{\sum_{i=x}^{y}\sum_{j=x}^{y} a_{ij}}{(y-x+1)^2} \tag{2}$$

Generally, the span score of a span is not only related to its adjacent span, but all other words outside the span. Therefore

[1] The negative syntactic distance is not a negative number, it is the number of common ancestors of two leaves. The smaller the syntactic distance is, the more ancestors the two leaves share.

we formulate the outside association as the average attention weights between the words inside the span and all other words outside the span:

$$s_{\text{out}}(x,y) = \frac{\sum\limits_{i=x}^{y}\sum\limits_{j=0,j\notin[x,y]}^{n-1} a_{ij} + \sum\limits_{i=0,i\notin[x,y]}^{n-1}\sum\limits_{j=x}^{y} a_{ij}}{2(y-x+1)n - 2(y-x+1)^2} \tag{3}$$

where n is the length of the sentence that consumes the span (x, y). The numerator of the above equation is the sum of all attention weights a_{ij} corresponding to the outside association. The first component in the numerator estimates the attention association from words inside the span to words outside the span while the second indicates the attention association from words outside the span to words inside the span.

Given the inside association score $s_{\text{in}}(x, y)$ and the outside association score $s_{\text{out}}(x, y)$, the score of span (x, y) to be a constituent can be estimated as follows:

$$s_{\text{span}}(x,y) = s_{\text{in}}(x,y) - s_{\text{out}}(x,y) \tag{4}$$

Given the span score of two adjacent spans: $s_{\text{span}}(x, z-1)$ and $s_{\text{span}}(z, y)$, we can estimate the split score as the sum of two span scores:

$$s_{\text{split}}(x,y,z) = s_{\text{span}}(x,z-1) + s_{\text{span}}(z,y) \tag{5}$$

The estimated split scores could be used for parsing with a greedy or chart-based algorithm.

3 Few-Shot Parsing

The attention weight matrix of BERT is the normalized dot product of a query $\mathbf{Q}$ and a key $\mathbf{K}$, as shown in Eq. (6). The essential information of the query and key is from the hidden representation $\mathbf{H}_{L-1}$ of the previous layer. They are computed as the product of $\mathbf{H}_{L-1}$ with $\mathbf{W}_Q$ and $\mathbf{W}_K$, as shown in Eq. (7). We denote the dimension of $\mathbf{H}_{L-1}$ as d_{model}. $\mathbf{W}_Q, \mathbf{W}_K \in R^{d_{\text{model}} \times d_{\text{model}}/h}$, h is the number of attention heads. BERT uses two matrices ($\mathbf{W}_Q, \mathbf{W}_K$) to linearly project $\mathbf{H}_{L-1}$ onto different sub-spaces and then computes the attention matrix as follows:

$$Attention = \text{softmax}(\frac{\mathbf{QK}^T}{\sqrt{d_{\text{model}}}}) \tag{6}$$

$$\begin{aligned} \mathbf{Q} &= \mathbf{H}_{L-1}\mathbf{W}_Q \\ \mathbf{K} &= \mathbf{H}_{L-1}\mathbf{W}_K \end{aligned} \tag{7}$$

We have explored all attention weight matrices of BERT to unsupervisedly parse sentences[2], and have found that the parsing results of different heads vary greatly even in the same layer. The differences among different heads are essentially from the different two linear projections: $\mathbf{W}_Q$ and $\mathbf{W}_K$. Therefore, we conjecture that they have a great impact on parsing and that we may infer better parse trees with better projections.

To verify this, we freeze other parameters of BERT, and retrain the two linear projections ($\mathbf{W}_Q, \mathbf{W}_K$) with annotated trees. Given the binarized tree of a sentence of length n, there are $n - 1$ split points in the tree, and each split point is corresponding to a span of the sentence and an internal node of the tree. We denote the jth split point as split_j and the span split by split_j as span_j. We can compute the split scores of these split points with Eq. (1) or Eq. (5). To make the computation suitable to our training, we normalize the split scores of each span using the softmax function:

$$p(\text{split}_j \mid \text{span}_j; \theta) = \frac{e^{s_{\text{split}}(\text{span}_j, \text{split}_j)}}{\sum_{k=1}^{n-1} e^{s_{\text{split}}(\text{span}_j, k)}} \tag{8}$$

where θ is the two matrices $\mathbf{W}_Q$, $\mathbf{W}_K$, $s_{\text{split}}(\text{span}_j, k)$ is defined in Eq. (1) or Eq. (5). The normalized scores can be seen as the probability of choosing a split point from the corresponding span. Multiplying the probabilities of all split points in a tree, we can get the probability of this tree. Given a binarized ground-truth tree, which contains $n - 1$ split points, we can compute the probability of the tree as follows:

$$p(\text{tree} \mid S; \theta) = \prod_{\text{split}_j \in \text{tree}} p(\text{split}_j \mid \text{span}_j; \theta) \tag{9}$$

where $\text{split}_j \in$ tree denotes that split_j is corresponding to a node in the tree and S represents the sentence. We maximize the probability of ground-truth trees to optimize θ, which is equivalent to minimizing the negative log likelihood:

$$L_\theta^{\text{MLE}} = -\sum_{i}^{N} log(p(\text{tree}_i \mid S_i; \theta)) \tag{10}$$

where N is the number of annotated trees used for training θ, tree_i is the binarized annotated tree of sentence S_i. Since the parameters to be tuned are just $\mathbf{W}_Q$ and $\mathbf{W}_K$, the two linear projection matrices, we can use a few tree samples to well train them. At inference, we use the trained projection matrices to replace the original ones in BERT.

4 Experiments

We evaluated our unsupervised and few-shot parsers on PTB [Dahlmeier *et al.*, 2013] for English and SPMRL [Seddah and others, 2013] for eight languages. For experiments on PTB, our models were built based on the pretrained base-uncased BERT released by Devlin *et al.* [2019]. While for experiments on SPMRL, our models were built based on the pretrained multilingual-base-uncased BERT released by Devlin *et al.* [2019]. We denote our unsupervised parser based on the syntactic distance (outside association) as **UPOA**, the unsupervised parser based on the strength of both the inside and outside association as **UPIO**, the few-shot parser based on the syntactic distance (outside association) as the **FPOA**, and the few-shot parser based on both the inside and outside association as the **FPIO**. The detailed experiment settings can be found in Zeng and Xiong [2022].

4.1 Unsupervised and Few-Shot Parsing on PTB

Unsupervised parsing. We compared UPOA and UPIO with the previous unsupervised parsing methods on PTB. The results are shown in Table 2. Although UPOA and UPIO

[2] The experiment results can be found in Section 5.2 of our full paper [Zeng and Xiong, 2022]

Model	Korean	German	Polish	Hungarian	Basque	French	Hebrew	Swedish
	Sentence-level F1							
Kim *et al.* [2020a]	**45.7**	39.3	42.3	38.0	**41.1**	**45.5**	42.8	38.7
UPOA	43.6	**40.9**	46.4	**38.5**	27.2	40.2	**44.5**	**42.0**
UPIO	40.3	40.0	**46.8**	35.5	24.7	38.8	42.8	41.9
	Corpus-level F1							
UPOA	**48.9**	41.59	49.71	42.3	36.43	38.63	45.08	42.97
UPIO	45.5	40.4	49.8	39.5	33.9	36.6	43.1	42.6
FPOA	46.4	**42.2**	60.2	43.7	29.5	50.5	60.1	57.8
FPIO	38.6	41.7	**60.3**	45.31	**30.9**	50.9	**60.6**	57.8
Berkeley	29.4	40.0	35.6	**50.3**	29.4	**57.0**	42.2	**68.6**
LB	25.6	16.5	25.4	18.3	29.2	9.0	12.1	15.7
RB	25.4	18.7	47.6	19.3	27.0	26.5	32.1	37.0

Table 1: F1 scores of models (tuned/trained on PTB) evaluated on 8 languages of SPMRL. The hyper-parameters of Kim *et al.* [2020b], UPOA and UPOA were tuned on the validation set of PTB, while the parameters of FPOA, FPIO and Berkeley parser were trained on 80 trees from the training set of PTB. The results of Kim *et al.* [2020b] are produced with their top-down parsing algorithm.

Model	Corpus level F1 score					
	WSJ-test			WSJ10		
	μ	σ	**max**	μ	σ	**max**
ON-LSTM	47.7	1.6	49.4	65.1	1.7	66.8
Tree-T	49.5	-	51.1	66.2	-	68.0
C-PCFG	**52.4**	-	-	-	-	-
UPOA	50.9	0.2	**51.4**	72.6	0.6	73.8
UPIO	50.3	0.4	51.0	**73.8**	0.4	74.5
Random	21.6	-	21.6	31.9	-	31.9
LB	9.0	-	9.0	19.6	-	19.6
RB	39.8	-	39.8	56.6	-	56.6

Table 2: F1 scores of unsupervised parsing on PTB. WSJ-test denotes the test set of PTB, while WSJ10 is a subset of the whole PTB where sentence length is ≤ 10. The results shown in the table are evaluated with corpus-level F1 score. ON-LSTM: Shen *et al.* [2019]. Tree-T: Tree-Transformer [Wang *et al.*, 2019]. C-PCFG: Kim *et al.* [2019a]

underperforms the state of the art (CPCFG), they are substantially better than the other models, especially on short sentences (WSJ10), which indicates that the self-attention weights of BERT can actually approximate the syntactic distances and be used to estimate the association scores.

Few-Shot parsing. We also compared FPOA and FPIO with a few-shot parsing method [Shi *et al.*, 2020] and the supervised state of the art (Berkeley parser [Kitaev *et al.*, 2019]) on PTB. The results are shown in Table 3. Training on a few (10,20,40) annotated trees, both FPIO and FPOA outperform the supervised Berkeley parser substantially and consistently. FPIO is also better than the compared few-shot parser [Shi *et al.*, 2020] training on 10 annotated trees.

4.2 Cross-Lingual Parsing on SPMRL

We evaluated the cross-lingual parsing performance of both unsupervised parsers (UPOA, UPIO) and few-shot parsers (FPOA, FPIO) on SPMRL datasets [Seddah and others, 2013]. We first tuned the hyper-parameters of (UPOA and UPIO) and the parameters of (FPOA and FPIO) on PTB, and then evaluated them on SPMRL dataset [Seddah and others, 2013]. We compared our models with a multilingual unsupervised parser [Kim *et al.*, 2020b], and the multilingual Berkeley parser [Kitaev *et al.*, 2019]. The detailed experiment settings can be found in our full paper [Zeng and Xiong, 2022]. The results are shown in Table 1. Both our unsupervised parsers (UPOA, UPIO) and few-shot parsers (FPOA, FPIO) significantly outperform the supervised Berkeley parser on 5 languages (8 languages in total). UPOA is also superior to the compared unsupervised parser [Kim *et al.*, 2020b] on another 5 languages.

Model	10		20		40	
	F1_{μ}	F1_{max}	F1_{μ}	F1_{max}	F1_{μ}	F1_{max}
FPOA	48.4	50.3	54.4	55.5	58.0	59.7
FPIO	**53.7**	**57.4**	**60.4**	**61.6**	**65.3**	**66.6**
Berkeley	27.2	32.8	38.3	40.3	47.8	49.9
FSS	-	53.4	-	-	-	-

Table 3: F1 scores of FPOA and FPIO trained on different numbers of annotated trees from the validation set of PTB, evaluated on the test data of PTB. FSS: a few-shot parsing method [Shi *et al.*, 2020]. Berkeley: a supervised parsing method [Kitaev *et al.*, 2019]

4.3 Analysis

More experiments and analyses can be found in our full paper [Zeng and Xiong, 2022], from which we have the following findings: 1) both UPOA and UPIO only need a few (even only 1) annotated trees for hyper-parameter tuning, 2) FPOA/FPIO achieves the best few-shot parsing performance on middle layers of BERT, 3) the linear projection matrices in FPOA/FPIO are in low-rank, 4) we can detect constituents from sentences by visualizing the attention weight matrices used for parsing.

References

[Dahlmeier *et al.*, 2013] Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. Building a large annotated corpus of learner english: The NUS corpus of learner english.

In Joel R. Tetreault, Jill Burstein, and Claudia Leacock, editors, *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@NAACL-HLT 2013, June 13, 2013, Atlanta, Georgia, USA*, pages 22–31. The Association for Computer Linguistics, 2013.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[Drozdov *et al.*, 2019] Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1129–1141. Association for Computational Linguistics, 2019.

[Jawahar *et al.*, 2019] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3651–3657. Association for Computational Linguistics, 2019.

[Kim *et al.*, 2019a] Yoon Kim, Chris Dyer, and Alexander M. Rush. Compound probabilistic context-free grammars for grammar induction. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2369–2385. Association for Computational Linguistics, 2019.

[Kim *et al.*, 2019b] Yoon Kim, Alexander M. Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. Unsupervised recurrent neural network grammars. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1105–1117. Association for Computational Linguistics, 2019.

[Kim *et al.*, 2020a] Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[Kim *et al.*, 2020b] Taeuk Kim, B. Li, and Sanggoo Lee. Chart-based zero-shot constituency parsing on multiple languages. *arXiv: Computation and Language*, 2020.

[Kitaev *et al.*, 2019] Nikita Kitaev, Steven Cao, and Dan Klein. Multilingual constituency parsing with self-attention and pre-training. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3499–3505. Association for Computational Linguistics, 2019.

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

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[Seddah and others, 2013] Djamé Seddah et al. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In Yoav Goldberg, Yuval Marton, Ines Rehbein, and Yannick Versley, editors, *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages, SPMRL@EMNLP 2013, Seattle, Washington, USA, October 18, 2013*, pages 146–182. Association for Computational Linguistics, 2013.

[Shen *et al.*, 2018a] Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron C. Courville. Neural language modeling by jointly learning syntax and lexicon. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[Shen *et al.*, 2018b] Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron C. Courville, and Yoshua Bengio. Straight to the tree: Constituency parsing with neural syntactic distance. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1171–1180. Association for Computational Linguistics, 2018.

[Shen *et al.*, 2019] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron C. Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[Shi *et al.*, 2020] Haoyue Shi, Karen Livescu, and Kevin Gimpel. On the role of supervision in unsupervised constituency parsing. In Bonnie Webber, Trevor Cohn, Yulan

He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7611–7621. Association for Computational Linguistics, 2020.

[Tenney *et al.*, 2019] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601. Association for Computational Linguistics, 2019.

[Wang *et al.*, 2019] Yau-Shian Wang, Hung-yi Lee, and Yun-Nung Chen. Tree transformer: Integrating tree structures into self-attention. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1061–1070. Association for Computational Linguistics, 2019.

[Zeng and Xiong, 2022] Zhiyuan Zeng and Deyi Xiong. Unsupervised and few-shot parsing from pretrained language models. *Artif. Intell.*, 305:103665, 2022.