Do Attention Heads in BERT Track Syntactic Dependencies?

Phu Mon Htut[*1] **Jason Phang**[*1] **Shikha Bordia**[*†2] **Samuel R. Bowman**[1,2,3]

[1]Center for Data Science
New York University
60 Fifth Avenue
New York, NY 10011

[2]Dept. of Computer Science
New York University
60 Fifth Avenue
New York, NY 10011

[3]Dept. of Linguistics
New York University
10 Washington Place
New York, NY 10003

Abstract

We investigate the extent to which individual attention heads in pretrained transformer language models, such as BERT and RoBERTa, implicitly capture syntactic dependency relations. We employ two methods—taking the maximum attention weight and computing the maximum spanning tree—to extract implicit dependency relations from the attention weights of each layer/head, and compare them to the ground-truth Universal Dependency (UD) trees. We show that, for some UD relation types, there exist heads that can recover the dependency type significantly better than baselines on parsed English text, suggesting that some self-attention heads act as a proxy for syntactic structure. We also analyze BERT fine-tuned on two datasets—the syntax-oriented CoLA and the semantics-oriented MNLI—to investigate whether fine-tuning affects the patterns of their self-attention, but we do not observe substantial differences in the overall dependency relations extracted using our methods. Our results suggest that these models have some specialist attention heads that track individual dependency types, but no generalist head that performs holistic parsing significantly better than a trivial baseline, and that analyzing attention weights directly may not reveal much of the syntactic knowledge that BERT-style models are known to learn.

1 Introduction

Pretrained Transformer models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have shown stellar performance on language understanding tasks, significantly improve the state-of-the-art on many tasks such as dependency parsing (Zhou et al., 2019), question answering (Rajpurkar et al., 2016), and have attained top positions on transfer learning benchmarks such as GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a). As these models become a staple component of many NLP tasks, it is crucial to understand what kind of knowledge they learn, and why and when they perform well. To that end, researchers have investigated the linguistic knowledge that these models learn by analyzing BERT (Goldberg, 2018; Lin et al., 2019) directly or training probing classifiers on the contextualized embeddings or attention heads of BERT (Tenney et al., 2019b,a; Hewitt and Manning, 2019).

BERT and RoBERTa, as Transformer models (Vaswani et al., 2017), compute the hidden representation of all the attention heads at each layer for each token by attending to all the token representations in the preceding layer. In this work, we investigate the hypothesis that BERT-style models use at least some of their attention heads to track syntactic dependency relationships between words. We use two dependency relation extraction methods to extract dependency relations from each self-attention heads of BERT and RoBERTa. The first method—maximum attention weight (MAX)—designates the word with the highest incoming attention weight as the parent, and is meant to identify specialist heads that track specific dependencies like `obj` (in the style of Clark et al., 2019). The second—maximum spanning tree (MST)—computes a maximum spanning tree over the attention matrix, and is meant to identify generalist heads that can form complete, syntactically informative dependency trees. We analyze the extracted dependency relations and trees to investigate whether the attention heads of these models track syntactic dependencies significantly better than chance or baselines, and what type of dependency relations they learn best. In contrast to probing models (Adi et al., 2017; Conneau et al.,

[*]Equal contribution

[†]Currently working at Verisk Analytics. This work was completed when the author was at New York University.

2018), our methods require no further training. In prior work, Clark et al. (2019) find that some heads of BERT exhibit the behavior of some dependency relation types, though they do not perform well at all types of relations in general. We are able to replicate their results on BERT using our MAX method. In addition, we also perform a similar analysis on BERT models fine-tuned on natural language understanding tasks as well as RoBERTa.

Our experiments suggest that there are particular attention heads of BERT and RoBERTa that encode certain dependency relation types such as `nsubj`, `obj` with substantially higher accuracy than our baselines—a randomly initialized Transformer and relative positional baselines. We find that fine-tuning BERT on the syntax-oriented CoLA does not significantly impact the accuracy of extracted dependency relations. However, when fine-tuned on the semantics-oriented MNLI dataset, we see improvements in accuracy for longer-distance clausal relations and a slight loss in accuracy for shorter-distance relations. Overall, while BERT and RoBERTa models obtain non-trivial accuracy for some dependency types such as `nsubj`, `obj` and `conj` when we analyze individual heads, their performance still leaves much to be desired. On the other hand, when we use the MST method to extract full trees from specific dependency heads, BERT and RoBERTa fail to meaningfully outperform our baselines. Although the attention heads of BERT and RoBERTa capture several specific dependency relation types somewhat well, they do not reflect the full extent of the significant amount of syntactic knowledge that these models are known to learn.

2 Related Work

Previous works have proposed methods for extracting dependency relations and trees from the attention heads of the transformer-based neural machine translation (NMT) models. In their preliminary work, Mareček and Rosa (2018) aggregate the attention weights across the self-attention layers and heads to form a single attention weight matrix. Using this matrix, they propose a method to extract constituency and (undirected) dependency trees by recursively splitting and constructing the maximum spanning trees respectively. In contrast, Raganato and Tiedemann (2018) train a Transformer-based machine translation model on different language pairs and extract the maximum spanning tree algorithm from the attention weights of the encoder for each layer and head individually. They find that the best dependency score is not significantly higher than a right-branching tree baseline. Voita et al. (2019) find the most confident attention heads of the Transformer NMT encoder based on a heuristic of the concentration of attention weights on a single token, and find that these heads mostly attend to relative positions, syntactic relations, and rare words.

Additionally, researchers have investigated the syntactic knowledge that BERT learns by analyzing the contextualized embeddings (Warstadt et al., 2019a) and attention heads of BERT (Clark et al., 2019). Goldberg (2018) analyzes the contextualized embeddings of BERT by computing language model surprisal on subject-verb agreement and shows that BERT learns significant knowledge of syntax. Tenney et al. (2019b) introduce a probing classifier for evaluating syntactic knowledge in BERT and show that BERT encodes syntax more than semantics. Hewitt and Manning (2019) train a *structural probing* model that maps the hidden representations of each token to an inner-product space that corresponds to syntax tree distance. They show that the learned spaces of strong models such as BERT and ELMo (Peters et al., 2018) are better for reconstructing dependency trees compared to baselines. Clark et al. (2019) train a probing classifier on the attention-heads of BERT and show that BERT's attention heads capture substantial syntactic information. While there has been prior work on analysis of the attention heads of BERT, we believe we are the first to analyze the dependency relations learned by the attention heads of fine-tuned BERT models and RoBERTa.

3 Methods

3.1 Models

BERT (Devlin et al., 2019) is a Transformer-based masked language model, pretrained on BooksCorpus (Zhu et al., 2015) and English Wikipedia, that has attained stellar performance on a variety of downstream NLP tasks. RoBERTa (Liu et al., 2019) adds several refinements to BERT while using the same model architecture and capacity, including a longer training schedule over more data, and shows significant improvements over BERT on a wide range of NLP tasks. We run our experiments on the pretrained large versions of both BERT (cased and uncased) and RoBERTa models, which consist of 24 self-attention layers with 16 heads each layer. For a given dataset, we feed each input sentence through the respective model and capture the attention weights for each individual head and layer.

Phang et al. (2018) report the performance gains on the GLUE benchmark by supplementing pretrained BERT with data-rich supervised tasks such as the Multi-Genre Natural Language Inference dataset (MNLI; Williams et al., 2018). Although these fine-tuned BERT models may learn different aspects of language and show different performance from BERT on GLUE benchmark, comparatively little previous work has investigated the syntax learned by these fine-tuned models (Warstadt et al., 2019a). We run experiments on the uncased BERT-large model fine-tuned on the Corpus of Linguistic Acceptability (CoLA; Warstadt et al., 2019b) and MNLI to investigate the impact of fine-tuning on a syntax-related task (CoLA) and a semantic-related task (MNLI) on the structure of attention weights and resultant extracted dependency relations. We refer to these fine-tuned models as *CoLA-BERT* and *MNLI-BERT* respectively.

3.2 Analysis Methods

We aim to test the hypothesis that the attention heads of BERT learn syntactic relations implicitly, and that the self-attention between two words corresponds to their dependency relation. We use two methods for extracting dependency relations from the attention heads of Transformer-based models. Both methods operate on the attention weight matrix $W \in (0,1)^{T \times T}$ for a given head at a given layer, where T is the number of tokens in the sequence, and the rows and columns correspond to the attending and attended tokens respectively (such that each row sums to 1).

Method 1: Maximum Attention Weights (MAX) Given a token A in a sentence, a self-attention mechanism is designed to assign high attention weights on tokens that have some kind of relationship with token A (Vaswani et al., 2017). Therefore, for a given token A, a token B that has the highest attention weight with respect to the token A should be related to token A. Our aim is to investigate whether this relation maps to a universal dependency relation. We assign a relation (w_i, w_j) between word w_i and w_j if $j = \text{argmax}\ W[i]$ for each row (that corresponds to a word in attention matrix) i in attention matrix W. Based on this simple strategy, we extract relations for all sentences in our evaluation datasets. This method is similar to Clark et al. (2019), and attempts to recover individual arcs between words; the relations extracted using this method need not form a valid tree, or even be fully connected, and the resulting edge directions may or may not match the canonical directions. Hence, we evaluate the resulting arcs individually and ignore their direction. After extracting dependency relations from all heads at all layers, we take the maximum UUAS over all relations types.

Method 2: Maximum Spanning Tree (MST) We also want to investigate if there are attention heads of BERT that can form complete, syntactically informative parse trees. To extract complete valid dependency trees from the attention weights for a given layer and head, we follow the approach of Raganato and Tiedemann (2018) and treat the matrix of attention weight tokens as a complete weighted directed graph, with the edges pointing from the output token to each attended token. As in Raganato and Tiedemann, we take the root of the gold dependency tree as the starting node and apply the Chu-Liu-Edmonds algorithm (Chu, 1965; Edmonds, 1967) to compute the maximum spanning tree. (Using the gold root as the starting point in MST may artificially improve our results slightly, but this bias is applied evenly across all the models we compare.) The resulting tree is a valid directed dependency tree, though we follow Hewitt and Manning (2019) in evaluating it as undirected, for easier comparison with our MAX method.

Following Voita et al. (2019), we exclude the sentence demarcation tokens (`[CLS]`, `[SEP]`, `<s>`, `</s>`) from the attention matrices. This allows us to focus on inter-word attention. Where the tokenization of our parsed corpus does not match the model's tokenization, we merge the non-matching tokens until the tokenizations are mutually compatible, and sum the attention weights for the corresponding columns and rows. We then apply either of the two extraction methods to the attention matrix. During evaluation when we compare the gold dependencies, to handle the subtokens within the merged tokens, we set all

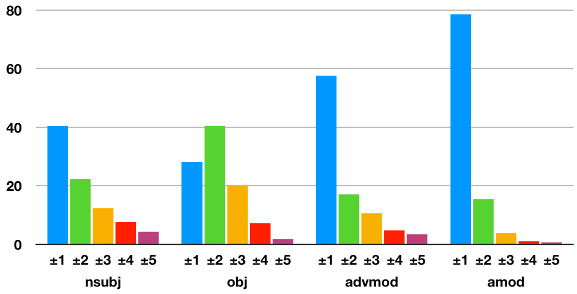

Figure 1: The distribution of the relative positions of `nsubj`, `obj`, `advmod`, `amod` in PUD.

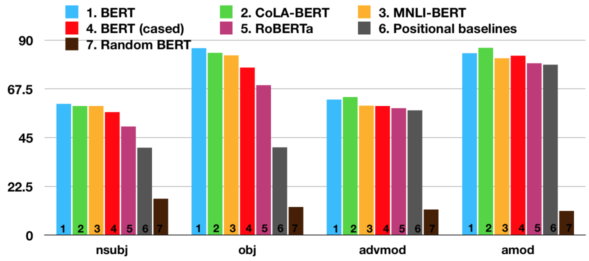

Figure 2: Undirected dependency accuracies by type based on our MAX method.

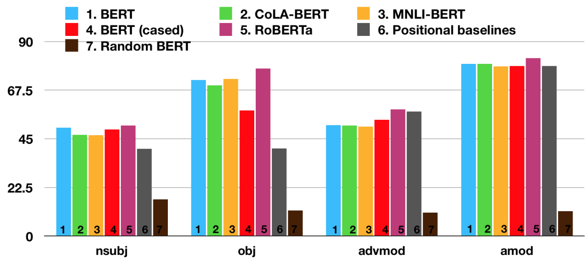

Figure 3: Undirected dependency accuracies by type based on the MST method.

subtokens except for the first to depend on the first subtoken. This approach is largely similar to that in Hewitt and Manning (2019). We use the English Parallel Universal Dependencies (PUD) treebank from the CoNLL 2017 shared task (Zeman et al., 2018) as the gold standard for our evaluation.

Baselines Many dependency relations tend to occur in specific positions relative to the parent word. For example, `amod` (adjectival modifier) mostly occurs before a noun. As an example, Figure 1 shows the distribution of relative positions for four major UD relations in our data. Following Voita et al. (2019), we compute the most common positional offset between a parent and child word for a given dependency relation, and formulate a baseline based on that most common relative positional offset to evaluate our methods. For MST, as we also want to evaluate the quality of the entire tree, we use a right-branching dependency tree as baseline. Additionally, we use a BERT-large model with randomly initialized weights (which we refer to as *random BERT*), as previous work has shown that randomly initialized sentence encoders perform surprisingly well on a suite of NLP tasks (Zhang and Bowman, 2018; Wieting and Kiela, 2019).

4 Results

Figure 2 and Table 1 describe the accuracy for the most frequent relation types in the dataset using relations extracted based on the MAX method. We also include results for the rarer long-distance `advcl` and `csubj` dependency types, as they show that MNLI-BERT has a tendency to track clausal dependencies better than BERT, CoLA-BERT, and RoBERTa. The non-random models outperform random BERT substantially for all dependency types. They also outperform the relative position baselines for some relation types. They outperform the baselines by a large margin for `nsubj` and `obj`, but only slightly better for `advmod` and `amod`. These results are consistent with the findings of Clark et al. (2019). Moreover, we do not observe substantial differences in accuracy by fine-tuning on CoLA. Both BERT and CoLA-BERT have similar or slightly better performance than MNLI-BERT, except for clausal dependencies such as `advcl` (adverbial clause modifier) and `csubj` (clausal subject) where MNLI-BERT outperforms BERT and CoLA-BERT by more than 5 absolute points in accuracy. This suggests that fine-tuning on a semantics-oriented task encourages effective long-distance dependencies, although it slightly degrades the performance in other shorter-distance dependency types.

Figure 3 shows the accuracy for `nsubj`, `obj`, `advmod`, and `amod` relations extracted based on the MST method. Similar to the MAX method, we choose the best accuracy for each relation type. We observe that the models outperform the baselines by a large margin for `nsubj` and `obj`. However, the models do not outperform the positional baseline for `advmod` and `amod`. Surprisingly, RoBERTa performs worse than other BERT models in all categories when the MAX method is used to extract the trees, but it outperforms all other models when the MST method is used.

Model	nsubj	obj	advmod	amod	case	det	obl	nmod	punct	aux	conj	cc	mark	advcl	csubj
BERT (cased)	*56.8*	*77.4*	*59.6*	*82.7*	*83.8*	*93.2*	*31.0*	*51.7*	*40.2*	*80.0*	*45.0*	***73.5***	***72.1***	*29.4*	*55.6*
BERT	***60.7***	***86.4***	***62.6***	*84.0*	*88.8*	***96.3***	*32.1*	***66.7***	*40.5*	*81.0*	*48.9*	*67.4*	***72.1***	*26.6*	*48.1*
CoLA-BERT	*59.7*	*84.2*	*63.8*	***86.4***	***89.5***	*96.2*	***33.8***	*66.2*	*41.1*	***82.2***	***50.6***	*68.5*	*70.8*	*28.0*	*51.9*
MNLI-BERT	*59.5*	*83.0*	*59.7*	*81.7*	*87.9*	*95.3*	*32.4*	*63.3*	***41.3***	*78.6*	*50.5*	*65.5*	*68.5*	***34.5***	***63.0***
RoBERTa	*50.2*	*69.3*	*58.5*	*79.3*	*75.6*	*74.4*	*26.2*	*47.4*	*37.4*	*75.7*	*44.6*	*69.0*	*63.1*	*23.2*	*44.4*
Positional	40.4	40.5	57.6	78.7	38.7	56.7	24.0	35.4	18.6	55.5	27.8	43.4	53.7	10.23	25.9
Random-BERT	16.8	12.9	11.8	11.1	13.7	12.6	13.5	13.8	12.6	16.3	18.9	20.9	12.8	13.3	22.2

Table 1: Highest accuracy for the most frequent dependency types. **Bold** marks the highest accuracy for each dependency type based on our MAX method. *Italics* marks accuracies that outperform our trivial baselines.

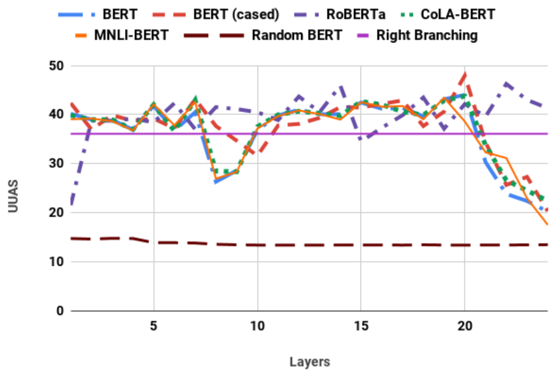

Figure 4: Maximum UUAS across layers of dependency trees extracted based on the MST method on PUD.

Figure 4 describes the maximum undirected unlabeled attachment scores (UUAS) across each layer. The trained models achieve significantly higher UUAS than random BERT. Although the trained models perform better than the right-branching baseline in most cases, the performance gap is not substantial. Given that the MST method uses the root of the gold trees, whereas the right-branching baseline does not, this implies that the attention weights in the different layers/heads of the BERT models do not appear to correspond to complete, syntactically informative parse trees.

Overall, the results of both analysis methods suggest that, although some attention heads of BERT capture specific dependency relation types, they do not reflect the full extent of the significant amount of syntactic knowledge BERT and RoBERTa are known to learn as shown in previous syntactic probing work (Tenney et al., 2019b; Hewitt and Manning, 2019). Additionally, we find that fine-tuning on the semantics-oriented MNLI dataset improves long term dependencies while slightly degrading the performance for other dependency types. The overall performance of BERT and the fine-tuned BERTs over the non-random baselines are not substantial, and fine-tuning on CoLA and MNLI also does not have a large impact on UUAS.

5 Conclusion

In this work, we investigate whether the attention heads of BERT and RoBERTa exhibit the implicit syntax dependency by extracting and analyzing the dependency relations from the attention heads at all layers. We use two simple dependency relation extraction methods that require no additional training, and observe that there are certain specialist attention heads of the models that track specific dependency types, but neither of our analysis methods support the existence of generalist heads that can perform holistic parsing. Furthermore, we observe that fine-tuning on CoLA and MNLI does not significantly change the overall pattern of self-attention within the frame of our analysis, despite their being tuned for starkly different downstream tasks.

Acknowledgments

This project grew out of a class project for the Spring 2019 NYU Linguistics seminar *Linguistic Knowledge in Reusable Sentence Encoders*. We are grateful to the department for making this seminar possible.

This material is based upon work supported by the National Science Foundation under Grant No. 1850208. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This project has also benefited from financial support to SB by Samsung Research under the project *Improving Deep Learning using Latent Structure* and from the donation of a Titan V GPU by NVIDIA Corporation.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Yoeng-Jin Chu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:13961400.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 19:233240.

Yoav Goldberg. 2018. Assessing BERT's syntactic abilities. Unpublished manuscript available on arXiv.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

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. Unpublished manuscript available on arXiv.

David Mareček and Rudolf Rosa. 2018. Extracting syntactic trees from transformer encoder self-attentions. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 347–349, Brussels, Belgium. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2018. Sentence encoders on STILTs: Supplementary training on intermediate labeled-data tasks. Unpublished manuscript available on arXiv.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the*

57th Annual Meeting of the Association for Computational Linguistics, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *Annual Conference on Neural Information Processing Systems*, abs/1905.00537.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019a. Investigating BERT's knowledge of language: Five analysis methods with NPIs. 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 2870–2880, Hong Kong, China. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019b. Neural network acceptability judgments. *TACL*, 7:625–641.

John Wieting and Douwe Kiela. 2019. No training required: Exploring random encoders for sentence classification. *International Conference on Learning Representations*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Zeman, Jan Hajic, 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, Brussels, Belgium, October 31 - November 1, 2018*, pages 1–21.

Kelly W. Zhang and Samuel R. Bowman. 2018. Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis. Unpublished manuscript available on arXiv.

Junru Zhou, Zhuosheng Zhang, and Hai Zhao. 2019. LIMIT-BERT : Linguistic informed multitask BERT. Unpublished manuscript available on arXiv.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. Unpublished manuscript available on arXiv.