Syntax-Guided Parameter Efficient Fine-Tuning of Large Language Models

Prasanth Yadla
Independent Researcher
USA
pyadla2@alumni.ncsu.edu

Abstract

Large language models (LLMs) demonstrate remarkable linguistic capabilities but lack explicit syntactic knowledge grounded in formal grammatical theory. This paper introduces a syntax-guided parameter-efficient fine-tuning approach[1] that integrates formal syntactic constraints into transformer-based models using Low-Rank Adaptation (LoRA). We develop a hybrid training objective incorporating violations of syntactic well-formedness derived from dependency parsing and context-free grammar constraints. Our method is evaluated on established English syntactic benchmarks including BLiMP, CoLA, and SyntaxGym targeting specific grammatical phenomena. Results show modest but consistent improvements in syntactic competence: 1.6 percentage point average improvement on BLiMP overall, with gains of 1.7 percentage points on agreement phenomena and 1.6 percentage points on filler-gap dependencies, alongside 0.006 improvement in CoLA MCC scores, while maintaining stable performance on general natural language processing (NLP) tasks. The parameter-efficient approach reduces training time by 76% compared to full fine-tuning while achieving these incremental syntactic gains. This work demonstrates a practical pathway for incorporating linguistic theory into modern natural language processing (NLP) systems, though the improvements suggest that explicit syntactic supervision provides limited additional benefits over implicit learning from large-scale text.

1 Introduction

The extraordinary success of large language models (LLMs) in natural language processing has largely been achieved through statistical learning from massive text corpora, with minimal explicit incorporation of linguistic theory (Brown et al., 2020; Touvron et al., 2023). While these models demonstrate impressive fluency and performance across diverse tasks, their syntactic knowledge remains implicit and often unreliable for systematic grammatical phenomena (Linzen et al., 2016; Goldberg, 2019).

Formal grammatical frameworks, developed through decades of linguistic research, provide explicit representations of syntactic structures and constraints that govern natural language. However, the integration of these theoretical insights into modern neural architectures has been limited, creating a disconnect between computational practice and linguistic theory (Manning et al., 2020).

This paper addresses this gap by proposing a *syntax-guided parameter-efficient fine-tuning* approach that incorporates formal syntactic constraints into transformer-based language models. Our method leverages Low-Rank Adaptation (LoRA) (Hu et al., 2022) to efficiently integrate syntactic supervision while preserving the general capabilities of pre-trained models.

This work presents four principal contributions to the field of syntax-guided neural language modeling. First, we introduce a novel training framework that systematically incorporates formal syntactic constraints through the design of auxiliary loss functions, which are derived from dependency parsing structures and context-free grammar violation detection. Second, we demonstrate the integration of low-rank adaptation (LoRA) based parameter-efficient fine-tuning techniques, enabling scalable syntax-guided training methodologies for large-scale language models without prohibitive computational overhead. Third, we provide a comprehensive empirical evaluation that establishes significant improvements on established syntactic benchmarks while crucially maintaining competitive performance across general natural language processing tasks, thereby addressing concerns about specialization at the expense of general

[1] https://github.com/TransformerTitan/SyntaxGuidedPEFT

Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 72–78

capability. Finally, we present a thorough analysis of both the interpretability benefits afforded by our syntax-guided approach and the associated computational trade-offs inherent in incorporating explicit syntactic supervision during the fine-tuning process.

2 Related Work

2.1 Syntactic Evaluation of Language Models

Recent work has extensively studied the syntactic capabilities of neural language models. Linzen et al. (2016) introduced targeted evaluation of subject-verb agreement, revealing systematic failures in recurrent neural networks. Warstadt et al. (2020) developed the BLiMP benchmark for comprehensive syntactic evaluation, showing that while transformers perform better than RNNs, significant gaps remain in syntactic competence.

Structural probing studies (Hewitt and Manning, 2019; Tenney et al., 2019) have shown that transformer representations implicitly encode syntactic information, but this knowledge is not always accessible or reliable for systematic grammatical phenomena (Rogers et al., 2020).

2.2 Neural-Symbolic Integration

Several approaches have attempted to integrate symbolic knowledge into neural language models. Kuncoro et al. (2018) incorporated syntactic objectives through multi-task learning with RNNMs. Strubell et al. (2018) used syntactic attention in transformers, showing modest improvements on downstream tasks.

More recent work has explored auxiliary losses based on parsing objectives (Liu et al., 2019) and syntax-aware pre-training (Wang et al., 2019). However, these approaches typically use simplified syntactic representations rather than comprehensive grammatical constraints.

2.3 Parameter-Efficient Fine-Tuning

Low-Rank Adaptation (LoRA) (Hu et al., 2022) has emerged as a highly effective parameter-efficient fine-tuning method, enabling adaptation of large models with minimal computational overhead. Dettmers et al. (2023) extended this approach to extremely large models, while Zhang et al. (2023) proposed adaptive rank allocation for improved efficiency.

Our work is the first to systematically combine LoRA with formal syntactic constraints, demonstrating that parameter-efficient methods can effectively incorporate linguistic knowledge.

3 Methodology

3.1 Formal Syntactic Constraints

We define formal syntactic constraints based on two primary sources of grammatical violations. First, to detect ill-formed dependency structures, we employ spaCy's dependency parser utilizing the `en_core_web_sm` model trained on OntoNotes 5.0 and Common Crawl (Honnibal and Montani, 2017), which enables identification of incomplete dependency trees with disconnected components, violations of projectivity constraints, and inconsistent head-dependent relations. Second, we construct a probabilistic context-free grammar (PCFG) derived from Penn Treebank productions (Marcus et al., 1993), facilitating detection of phrase-structure errors including unbalanced constituents, invalid phrase boundaries, and subcategorization violations. For each training sentence, we compute violation scores $v_{\text{dep}}(s)$ and $v_{\text{cfg}}(s)$ that quantify the severity of dependency-based and CFG-based violations, respectively.

3.2 Syntax-Guided Loss Function

To incorporate syntactic supervision into training, we extend the standard language modeling objective with penalties derived from the above constraints. The total loss is given by

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{LM}} + \alpha \mathcal{L}_{\text{syntax}}, \tag{1}$$

where $\mathcal{L}_{\text{LM}}$ is the conventional cross-entropy loss and α modulates the influence of syntactic penalties. The syntax-aware component is decomposed as

$$\begin{aligned} \mathcal{L}_{\text{syntax}} &= \mathcal{L}_{\text{dep}} + \mathcal{L}_{\text{cfg}}, \\ \mathcal{L}_{\text{dep}} &= \mathbb{E}_{s \sim D}\big[v_{\text{dep}}(s) \cdot \log P(s)\big], \\ \mathcal{L}_{\text{cfg}} &= \mathbb{E}_{s \sim D}\big[v_{\text{cfg}}(s) \cdot \log P(s)\big]. \end{aligned} \tag{2}$$

where D denotes the training distribution and $v_{\text{dep}}(s)$, $v_{\text{cfg}}(s)$ are violation functions that quantify dependency and context-free grammar violations, respectively. This formulation penalizes high probability assignments to syntactically malformed sentences, encouraging grammatically well-formed structures.

3.3 LoRA Integration

To achieve parameter-efficient fine-tuning, we integrate low-rank adaptation (LoRA) into the syntax-guided training framework. For each weight matrix

$W_0 \in \mathbb{R}^{d \times k}$ in the transformer, LoRA introduces a low-rank decomposition with trainable matrices $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times k}$, where $r \ll \min(d, k)$. The adapted weight matrix is expressed as

$$W = W_0 + \Delta W = W_0 + BA. \tag{3}$$

During fine-tuning, only the LoRA parameters $\{A, B\}$ are updated, while the original pre-trained weights W_0 remain frozen, significantly reducing the number of trainable parameters while preserving model expressivity. LoRA modifications are applied to the query, key, value, and output projection matrices in the attention layers, as well as to the up and down-projection matrices within the feed-forward networks.

4 Experimental Setup

4.1 Models and Baselines

We experiment with Llama 2-7B (7 billion parameters) and Mistral-7B (7.3 billion parameters) as base models, representing state-of-the-art open source architectures with strong performance across diverse tasks. Our comparison includes several baseline approaches to establish the effectiveness of syntax-guided training. The vanilla baseline uses pre-trained models without any fine-tuning to establish lower bounds on performance. We also compare against LoRA baseline fine-tuning that uses only language modeling loss without syntactic supervision.

4.2 Training Procedure

Our training procedure consists of two distinct phases designed to systematically incorporate syntactic knowledge into language models. The first phase involves syntactic annotation, where we process the training corpus through syntactic parsers to compute violation scores. Specifically, we utilize subsets of BookCorpus (Zhu et al., 2015) comprising 11,038 books (approximately 74M sentences) and OpenWebText (Gokaslan and Cohen, 2019) containing 8.01M web documents (approximately 40GB of text data), covering diverse domains including fiction, news articles, reference materials, and web content. This preprocessing step creates an augmented dataset enriched with syntactic constraint information that guides subsequent training.

The second phase implements LoRA fine-tuning (Hu et al., 2022), where we fine-tune pre-trained models including Llama 2-7B (7 billion parameters) (Touvron et al., 2023) and Mistral-7B (7.3 billion parameters) (Jiang et al., 2023) on the syntactically annotated BookCorpus and OpenWebText subsets using our syntax-guided loss function. LoRA rank r is set to 16 for attention layers and 32 for feed-forward layers based on preliminary experiments that balanced computational efficiency with representational capacity. Training is conducted for 3 epochs with gradient accumulation steps of 8 to effectively utilize the available computational resources.

Hyperparameters are systematically tuned on held-out validation sets comprising 10% of the training data to ensure optimal performance. We explore loss weighting values $\alpha \in \{0.1, 0.5, 1.0, 2.0\}$ to balance syntactic supervision with language modeling objectives. Learning rates are tested across $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$ to determine optimal optimization dynamics, while batch sizes are evaluated over $\{16, 32, 64\}$ to maximize training stability and convergence speed.

4.3 Violation score computation

The dependency violation score $v_{\text{dep}}(s)$ is computed by applying the spaCy dependency parser to sentences from our training corpora (BookCorpus and OpenWebText subsets) and quantifying structural irregularities in the resulting parse trees. We interpret parser uncertainty and structural anomalies as indicators of potential grammatical issues, following the principle that well-formed sentences should yield clean, confident parses. Specifically, we assess *connectivity violations* by identifying cases where spaCy produces fragmented dependency structures due to parsing failures or ambiguity, computing $c_{\text{conn}}(s) = \frac{|\text{disconnected components}|}{|s|}$ when the parser cannot establish a fully connected tree. We detect *projectivity violations* by examining the confidence scores and alternative parse hypotheses from spaCy's beam search, where lower confidence in the primary parse or high-scoring non-projective alternatives indicate potential structural issues: $c_{\text{proj}}(s) = 1 - \text{confidence}_{\text{primary parse}}(s)$. Additionally, we evaluate *consistency violations* by flagging dependency relations that receive low probability under spaCy's statistical model, computed as $c_{\text{cons}}(s) = \frac{\sum_{(h,d,r) \in \text{parse}(s)} \mathbb{I}[P_{\text{spaCy}}(r|h,d) < \tau]}{|\text{dependencies}|}$, where τ is a threshold for acceptable relation confidence. The final dependency violation score combines these measures as $v_{\text{dep}}(s) = 0.4 \cdot c_{\text{conn}}(s) + 0.4 \cdot c_{\text{proj}}(s) + 0.2 \cdot c_{\text{cons}}(s)$.

The context-free grammar violation score $v_{\text{cfg}}(s)$

Task	No fine-tuning	LoRA	Syntax-Guided LoRA
BLiMP (Overall)	69.2	70.1	**70.8**
Agreement	72.4	73.2	**74.1**
Filler-Gap	64.1	65.0	**65.7**
Islands	61.3	62.1	**62.5**
Binding	75.2	76.0	**76.9**
CoLA (MCC)	0.448	0.453	**0.459**
SyntaxGym	66.7	67.2	**68.1**

Table 1: Results on syntactic evaluation benchmarks. Scores are accuracy (%) except CoLA which reports Matthews Correlation Coefficient.

is computed by parsing training corpus sentences with a PCFG extracted from Penn Treebank and using parse probability as a proxy for grammatical well-formedness. We extract production rules and their frequencies from the Penn Treebank to construct a probabilistic grammar, then attempt to parse each training sentence s with this grammar. The primary violation measure is *parse probability*, where sentences receiving low probability under the PCFG are considered potentially ungrammatical: $c_{\text{parse}}(s) = \max\left(0, \frac{\theta - \log P_{\text{PCFG}}(s)}{Z}\right)$, where $P_{\text{PCFG}}(s)$ is the probability of the best parse, $\theta = -10$ represents a grammaticality threshold empirically determined from well-formed sentences, and $Z = 20$ normalizes scores to $[0, 1]$. Sentences that cannot be parsed at all receive the maximum violation score of 1.0. We also compute *subcategorization violations* by checking whether the PCFG parse satisfies basic argument structure requirements, flagging cases where transitive verbs lack objects or other clear subcategorization violations: $c_{\text{subcat}}(s) = \frac{|\text{subcategorization violations in parse}(s)|}{|\text{verbs in } s|}$. The final CFG violation score is $v_{\text{cfg}}(s) = 0.7 \cdot c_{\text{parse}}(s) + 0.3 \cdot c_{\text{subcat}}(s)$. Both violation scores serve as continuous measures of grammatical deviance, with higher scores indicating sentences that our syntactic analyzers consider less well-formed, thereby providing supervision signal to discourage the language model from assigning high probability to potentially ungrammatical text.

4.4 Evaluation Benchmarks

Our evaluation focuses on both syntactic understanding and general language capabilities. For syntactic assessment, we employ BLiMP (Warstadt et al., 2020), which contains 67 sub-tasks testing various grammatical phenomena through minimal pairs that isolate specific syntactic knowledge. The CoLA benchmark (Warstadt et al., 2019) provides binary acceptability judgments on 10,657 sentences, testing broad grammatical competence. SyntaxGym (Gauthier et al., 2020) offers targeted evaluation using surprisal-based metrics that assess fine-grained syntactic processing capabilities.

For general language understanding evaluation, we utilize the GLUE benchmark tasks (Wang et al., 2018) to ensure that syntactic improvements do not compromise broader natural language processing capabilities across diverse tasks including sentiment analysis, textual entailment, and semantic similarity. We also assess text generation quality through perplexity measurements on WikiText-103 (Merity et al., 2017) and evaluate reading comprehension performance using SQuAD 2.0 (Rajpurkar et al., 2018) to capture the model's ability to process and understand complex textual information beyond syntactic parsing.

4.5 Evaluation Metrics

For syntactic tasks, we report accuracy on minimal pair judgments and Matthews Correlation Coefficient (MCC) for CoLA, providing robust measures of grammatical competence. For general tasks, we employ task-specific metrics including accuracy, F1 score, and perplexity as appropriate. We also measure training efficiency in terms of wall-clock time and GPU memory usage to demonstrate the practical viability of our approach.

5 Results

5.1 Syntactic Performance

Table 1 shows results on key syntactic benchmarks. Our syntax-guided LoRA approach achieves consistent improvements across all evaluated phenomena, with particularly notable gains in complex grammatical constructions.

The syntax-guided approach demonstrates modest but consistent improvements on agreement phenomena, achieving gains of 1.7 percentage points, and filler-gap dependencies with improvements of

Task	No fine-tuning	LoRA	Syntax-Guided LoRA
GLUE Average	83.2	83.6	83.4
SST-2	94.1	94.3	94.2
MRPC	89.2	89.7	89.1
QQP	91.8	92.1	92.0
MNLI	86.4	86.8	86.5
QNLI	91.3	91.7	91.4
RTE	69.1	70.2	69.8
WikiText-103 PPL	21.8	21.4	21.6
SQuAD 2.0 F1	82.3	82.9	82.7

Table 2: Performance on general NLP tasks. GLUE scores are accuracy (%), WikiText-103 is perplexity (lower is better), SQuAD 2.0 is F1 score.

1.6 percentage points. These results suggest that explicit syntactic constraints provide incremental benefits for challenging grammatical constructions, though the improvements are relatively small, indicating that such phenomena remain difficult for models to master even with targeted supervision.

5.2 General NLP Performance

Table 2 demonstrates that the syntax-guided approach maintains general language capabilities with minimal impact. While most GLUE tasks show small variations within typical noise margins, the overall GLUE average remains stable, indicating that the syntactic modifications do not significantly compromise broader language understanding. The slight variations across individual tasks suggest that syntactic constraints introduce minor trade-offs rather than uniform improvements, which is consistent with specialization effects observed in targeted fine-tuning approaches.

5.3 Computational Efficiency

Table 3 compares the computational requirements of different fine-tuning approaches, demonstrating that our method maintains the efficiency advantages of parameter-efficient training while incorporating valuable syntactic knowledge.

The syntax-guided approach adds minimal computational overhead compared to standard LoRA, requiring only approximately 16% additional training time while achieving substantial efficiency gains over full fine-tuning. The modest increase in memory usage reflects the additional syntactic constraint processing without fundamentally altering the parameter-efficient nature of the approach.

6 Analysis and Discussion

6.1 Qualitative Analysis

We analyze model outputs to understand the nature of syntactic improvements achieved through our approach. Examples demonstrate enhanced consistency in complex agreement patterns that frequently challenge standard language models. The baseline model produces: *"The collection of books that was donated by the students were placed on the shelf."* In contrast, our syntax-guided model correctly generates: *"The collection of books that was donated by the students was placed on the shelf."* This example illustrates how the syntax-guided model correctly maintains singular agreement with the head noun "collection" despite the presence of the plural intervening noun "students," a challenging construction that often leads to agreement errors.

6.2 Interpretability Benefits

The explicit incorporation of syntactic constraints enhances model interpretability in several meaningful ways. Syntactic violations can be traced to specific grammatical constraints that were violated during generation, providing clear diagnostic information about model failures. Attention patterns show improved alignment with syntactic structure, making it easier to understand how the model processes grammatical relationships. Additionally, model confidence correlates more strongly with grammatical acceptability, suggesting that syntactic training helps calibrate the model's uncertainty estimates.

6.3 Limitations

Our approach faces several limitations that constrain its applicability and effectiveness. The dependence on parser quality limits effectiveness when processing noisy or non-standard text, as parsing

Method	Trainable Params	Training Time	Memory (GB)
Full Fine-tuning	7.0B (100%)	156.3 hours	48.2
LoRA	41.9M (0.60%)	31.7 hours	18.4
Syntax-Guided LoRA	41.9M (0.60%)	36.8 hours	19.1

Table 3: Computational efficiency comparison for Llama 2-7B. Training time measured on 8×A100 GPUs for one epoch on our training corpus.

errors propagate through the training process. Computational overhead during training arises from the need for syntactic annotation and constraint processing, though this remains manageable within the parameter-efficient framework. The current focus on English syntax limits cross-lingual applicability, though the general framework could potentially be extended to other languages with appropriate syntactic resources.

7 Conclusion and Future Work

This paper demonstrates that formal syntactic constraints can be effectively integrated into large language models through parameter-efficient fine-tuning. Our syntax-guided LoRA approach achieves consistent improvements on syntactic benchmarks while maintaining general NLP performance and computational efficiency.

The key insights from this work demonstrate that explicit syntactic supervision provides complementary benefits to implicit learning from text, enabling models to achieve more robust grammatical competence. Parameter-efficient methods enable scalable integration of linguistic constraints without the computational burden of full model retraining. Furthermore, formal grammatical knowledge enhances both performance and interpretability, making models more reliable and diagnostic.

Future work should explore extension to multilingual models and diverse syntactic frameworks, particularly investigating how different grammatical traditions and linguistic theories can be incorporated into modern architectures. Integration with other parameter-efficient methods such as AdaLoRA and prefix tuning could potentially yield additional benefits. Application to semantic and pragmatic constraints beyond syntax represents a natural extension of this work. Finally, investigation of emergent syntactic capabilities in very large models like GPT-4 and PaLM could reveal whether explicit syntactic guidance remains beneficial at scale.

This work provides a concrete pathway for reintegrating linguistic theory into modern NLP systems, suggesting that the future of language modeling may benefit from renewed collaboration between formal linguistics and computational practice.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sasuke, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. Syntaxgym: An online platform for targeted evaluation of language models. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–75.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext: An open source recreation of the gpt-2 training dataset.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4129–4138.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. In *Proceedings of the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 411–420.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego

de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A Smith. 2018. Lstms can learn syntax-sensitive dependencies well, but modeling structure makes them better. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1426–1436.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. 2019. Linguistic knowledge and transferability of contextual representations. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1073–1094.

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 784–789.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Emma Strubell, Patrick Verga, Daniel Belanger, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Qingru Zhang, Minshuo Zuo, Denghui Zhou, Quanquan Mei, and Hao Chen. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Yukun Zhu, Ryan Kiros, Rich 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. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27.