# Mixture-of-Linguistic-Experts Adapters for Improving and Interpreting Pre-trained Language Models

**Raymond Li**[†], **Gabriel Murray**[‡], **Giuseppe Carenini**[†]
[†]University of British Columbia, Vancouver, BC, Canada
[‡]University of Fraser Valley, Abbotsford, BC, Canada
{raymondl, carenini}@cs.ubc.ca
gabriel.murray@ufv.ca

## Abstract

In this work, we propose a method that combines two popular research areas by injecting linguistic structures into pre-trained language models in the parameter-efficient fine-tuning (PEFT) setting. In our approach, parallel adapter modules encoding different linguistic structures are combined using a novel Mixture-of-Linguistic-Experts architecture, where Gumbel-Softmax gates are used to determine the importance of these modules at each layer of the model. To reduce the number of parameters, we first train the model for a fixed small number of steps before pruning the experts based on their importance scores. Our experiment results with three different pre-trained models show that our approach can outperform state-of-the-art PEFT methods with a comparable number of parameters. In addition, we provide additional analysis to examine the experts selected by each model at each layer to provide insights for future studies.

## 1 Introduction

In recent years, pre-trained language models have become the de facto instrument for the field of natural language processing (NLP) (Devlin et al., 2019; Liu et al., 2019; Clark et al., 2020; He et al., 2021, 2023). This shift is largely due to the emergence and success of transformer-based models (Vaswani et al., 2017) where large-scale pre-training helps the model to learn the syntactic and semantic structure of a language without explicit supervision. At the same time, there are good reasons to question whether these models can be said to understand a language in any meaningful and interpretable way (Trott et al., 2020; Merrill et al., 2021). To address this conundrum, probing studies have demonstrated, to a certain extent, that it is possible to infer linguistic structures from the representations within these models (Hewitt and Manning, 2019; Tenney et al., 2019b; Maudslay et al., 2020). However, the precise connection between the existence of structures and their benefits to task performance is yet to be firmly established. On the other hand, while the conventional way of fine-tuning has found success in a wide array of NLP tasks, its applicability has increasingly diminished due to the associated computational expense with the recent shift towards larger and more complex models (Zhao et al., 2023).

While some have argued that pre-training on unstructured text alone equips the model with sufficient capacity to comprehend the meaning of language, others (Bender and Koller, 2020; Prange et al., 2022) have asserted that mapping the model's behavior onto human-comprehensible structures offers more dependable evidence of its ability to tackle tasks beyond merely exploiting superficial cues. Specifically, studies in this area have yielded successful attempts to inject syntactic and semantic structures into the pre-trained language models (Bai et al., 2021; Wu et al., 2021; Yu et al., 2022), with positive results reported on downstream tasks. However, despite recent efforts, no existing work has addressed the problem of where and how to effectively inject multiple different structures in an efficient manner.

The conventional approach of fine-tuning pre-trained NLP models involves optimizing the full set of model parameters for each task. However, this results in a separate copy of fine-tuned model parameters for each task and has become increasingly infeasible due to the recent trend of pre-training larger and larger models. To address these concerns, a surge of recent work has been dedicated to the study of parameter-efficient fine-tuning (PEFT) methods (Ding et al., 2023), where only a small portion of task-specific trainable parameters are tuned while keeping the rest of the model frozen. While these studies have achieved impressive performance even comparable to the full fine-tuning, they have been mostly focused on either determining the subset of model parameters for tuning (Lee et al., 2019; Ben Zaken et al., 2022) or finding the location to insert additional trainable parameters (Houlsby et al., 2019a; Li and Liang, 2021; Hu et al., 2022). No existing work has addressed the problem of whether linguistic structural priors can be incorporated into these trainable parameters under the PEFT setting.

In this work, we align the two research areas of injecting linguistic structures and PEFT by proposing a strategy of effectively combining multiple linguistic structures into pre-trained NLP models in a parameter-efficient fashion. To combine multiple linguistic structures, we propose a novel architecture inspired by Mixture-of-Experts models (Shazeer et al., 2017), where Relational Graph Convolutional Networks (RGCN) modules (Schlichtkrull et al., 2018) encoded with different linguistic trees are aggregated using learnable Gumbel-Softmax (Jang et al., 2017) gates, and inserted between each layer of the pre-trained model. To reduce the number of parameters, we propose a pruning strategy where we first tune the full set of RGCN modules before pruning all but the top "experts" based on the importance score learned from the gates. To demonstrate the benefits of our approach, we perform experiments on the GLUE benchmark with three different pre-trained NLP models and compare the results with state-of-the-art PEFT methods (Mao et al., 2022). Further, we perform additional analysis to understand which types of linguistic structures are kept at each layer of the model and provide insights for future work on injecting knowledge through PEFT methods. In short, our contributions can be summarized as the following:

1. We propose a novel architecture to effectively combine and interpret multiple linguistic structures at different layers of the pre-trained model.

2. To improve efficiency, we adopt a pruning strategy by keeping only the top experts according to their importance scores.

3. Our experimental results with three different models demonstrate the benefits of our approach by achieving the best overall performance on the GLUE benchmark.

4. We perform analysis on the experts selected by the model to providing valuable insights for future work.

## 2 Related Works

We organize this section based on the two research areas that our work seeks to align. In §2.1, we provide an overview of techniques to inject linguistic structure, while §2.2 summarizes recent trends in parameter-efficient fine-tuning.

### 2.1 Injecting Linguistic Structures

Earlier works on injecting linguistic structures into neural networks are often based on the recursive neural network architecture (Goller and Kuchler, 1996; Socher et al., 2011, 2012, 2013), where a compositional function recursively combines representations of child nodes following a predefined tree structure. Following the same intuition, subsequent studies have extended their approach for composing hidden states into a variety of neural architectures including recurrent neural networks (RNNs) (Tai et al., 2015; Miwa and Bansal, 2016; Roth and Lapata, 2016; Kuncoro et al., 2017; Shen et al., 2019), graph neural networks (GNNs) (Marcheggiani and Titov, 2017; Bastings et al., 2017; Zhang et al., 2018; Huang and Carley, 2019; Wang et al., 2020), and later, Transformers (Wu et al., 2018; Hao et al., 2019; Strubell et al., 2018; Wang et al., 2019b,c). For instance, Strubell et al. (2018) used the bi-affine operator (Dozat and Manning, 2017) to predict the affinity score between the token representations (key and query vector) based on the dependency tree, while (Wang et al., 2019c) encouraged the attention heads to follow tree structures by applying a constituent prior on the attention weights.

More recently, research in this area has shifted towards pre-trained language models (Devlin et al., 2019; Liu et al., 2019; Clark et al., 2020; He et al., 2021, 2023). While prior studies on probing (Hewitt and Manning, 2019; Tenney et al., 2019b; Maudslay et al., 2020; Newman et al., 2021; Arps et al., 2022) have shown that meaningful hierarchical structures (e.g., syntactic trees) can be extracted from pre-trained models without explicit supervision, it has also been found that incorporating linguistic structures can still be beneficial for downstream performance (Zhang et al., 2020; Kuncoro et al., 2020; Sachan et al., 2021; Qian et al., 2021), even when the structures already exist in the model (Li et al., 2022). For example, Bai et al. (2021) explicitly masked the existing pre-trained attention weights based on the adjacency matrices defined by the syntactic trees. On the other hand, Wu et al. (2021) used additional GNN layers to incorporate semantic dependencies by appending them on top of the pre-trained encoder. Most similar to our work, Yu et al. (2022) extended the approach by Wu et al. (2021) and performed an empirical study on syntax and trivial graphs. However, their method requires training a new model for each graph, which is inefficient for studying their benefits at different layers of the model. To the best of our knowledge, no existing works have attempted to incorporate multiple different linguistic structures within the same model, as we do in this paper.

### 2.2 Parameter Efficient Fine-tuning

While the standard paradigm of fine-tuning pre-trained language models has emerged as a common practice for NLP tasks (Min et al., 2021), it has become less applicable due to the computational cost associated with the increasingly large models (Brown et al., 2020a; OpenAI, 2023). Parameter-efficient fine-tuning (PEFT) methods (Ding et al., 2023), on the other hand, present a solution to this problem by freezing most or all of the pre-trained weights and only fine-tuning a small set of parameters in proportion to the model size. PEFT methods can be roughly organized into two categories. The first category tunes a subset of existing parameters with notable examples including freezing entire layers (Lee et al., 2019) or tuning only the bias terms (Ben Zaken et al., 2022). However, these approaches generally lead to worse performance and have only been shown to achieve comparable performance to full fine-tuning on low-resource tasks. Alternatively, the second category adds new trainable parameters while keeping the pre-trained weights frozen (Han et al., 2021; Karimi Mahabadi et al., 2021; Lester et al., 2021). For example, Houlsby et al. (2019a) used a trainable bottleneck layer after the feed-forward network in each layer of the model, Li and Liang (2021) prepended trainable vectors to the input of multi-head attention, while Hu et al. (2022) combined the pre-trained attention weights with trainable low-rank matrices. Lastly, more recent studies (He et al., 2022; Mao et al., 2022) proposed a unified framework by combining different PEFT methods as sub-modules. While we use their approach as our baselines, no existing PEFT works have attempted to incorporate interpretable structures as priors to the trainable modules, as we do in this paper.

## 3 Model Architecture

In this section, we describe the architectures of our Mixture-of-Linguistic adapters (Figure 1). We start by first introducing the Relational Graph Convolutional Network (RGCN) modules for incorporating linguistic structures (§3.1) before describing the method used for combining multiple RGCN (§3.2). Finally, we discuss how the adapters are inserted into the pre-trained model (§3.3).

### 3.1 Modeling Dependency Structures

To model dependency structures, we adopt the method proposed by Wu et al. (2021), where RGCN (Schlichtkrull et al., 2018) layers are used to propagate node representations according to the structure defined by the dependency tree.

$$h_i^{(\ell)} = \text{ReLU}\left( \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i} \frac{W_r h_j^{(\ell-1)}}{|\mathcal{N}_i|} + W_0 h_i^{(\ell-1)} \right) \tag{1}$$

Equation 1 describe the propagation process for a single RGCN layer, where the node representation $h_i$ is updated with a learned composition function based on the node's neighbors $h_j \in \mathcal{N}_i$ (and itself) in the dependency graph. Specifically, we use the intermediate hidden states of the pre-trained model as input, where sub-word token vectors are mean-pooled to create the node representation for the associated word. Since the number of parameters in RGCN linearly increases with the number of relation types, rather than associating each dependency relation with a separate set of weights $W_r$, we only model the child and parent relations ($|\mathcal{R}| = 2$) to reduce parameter count.

The graph convolution operation has a computational complexity $\mathcal{O}(|E| \cdot d_1 \cdot d_2)$, where $d_1$ and $d_2$ are respectively the number of input and output dimensions of the layer, and $|E|$ is the total number of edges defined by the dependency graph. In addition, the self-loop operation in the RGCN layer adds a complexity $\mathcal{O}(|N| \cdot d_1 \cdot d_2)$, where $|N| = |E| + 1$ is the total number of nodes or word tokens in the dependency graph. The self-loop operation has the same complexity as the standard linear layer.

### 3.2 Combining Different Modules

Inspired by the Mixture-of-Experts architecture (Shazeer et al., 2017), we propose a strategy

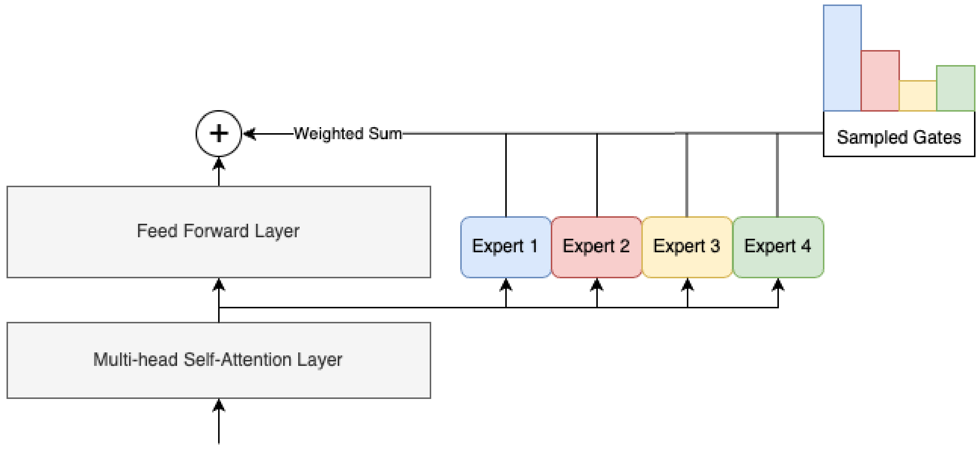


Figure 1: Our proposed Mixture-of-Linguistic-Experts architecture for a single transformer layer. In the four-expert configuration provided by the example, where the outputs from the expert modules are aggregated based on the weights sampled from the Gumbel-Softmax distribution.

to determine the importance of different adapter modules by sampling gate values from a Gumbel-Softmax distribution (Maddison et al., 2017; Jang et al., 2017). Specifically, we define a gate logit $z_i$ for each "expert" module $E_i$, where the gate value $g_i$ is sampled from the Gumbel-Softmax distribution during training. The sampling method is defined as:

$$g_i = \text{softmax}(z_i + \epsilon)/\tau \tag{2}$$

where the stochasticity comes from the gumbel noise $\epsilon = -\log(-\log(u))$ s.t. $u \sim \text{Uniform}(0, 1)$, and $\tau$ is the temperature to control the randomness of the distribution. The value of the gate logit $z_i$ can be interpreted as the contribution of the respective expert module when computing the aggregated representation from all experts.

In contrast to the softmax gates used in the original MoE architecture (Shazeer et al., 2017), sampling gates from the Gumbel-Softmax distribution provides more interpretability regarding its importance since the inherent stochasticity introduces an exploratory characteristic and allows the model to consider a diverse set of potential outputs. Meanwhile, the standard softmax operation assumes a single correct answer at each layer, meaning the model could possibly overlook good combinations of modules by locally exploiting the module with the single highest probability.

### 3.3 Adapters

Based on prior works on parameter efficient fine-tuning (Houlsby et al., 2019b; Mao et al., 2022), we inject our Mixture-Linguistic-Experts layer between the layers of pre-trained transformer model (§3.2) and update only the adapters while keeping the pre-trained parameters frozen. We choose to insert modules following the suggestions by He et al. (2022), where they found that inserting adapters in parallel to the feed-forward networks (FFN) achieves the overall best performance.

$$\begin{aligned} h_{\text{attn}}^{(\ell)} &= \text{MultiHeadAttn}(h^{(\ell-1)}) \\ h^{(\ell)} &= \text{FFN}(h_{\text{attn}}^{(\ell)}) + \text{Adapter}(h_{\text{attn}}^{(\ell)}) \end{aligned} \tag{3}$$

From Equation 3, our adapter module takes the hidden output $h_{\text{attn}}$ from the multi-head attention (MultiHeadAttn) sub-layer and uses an additive composition with the original FFN to create the final layer output $h^{(l)}$.

## 4 Training Strategy

While the architecture proposed in section 3 allows us to aggregate multiple adapter modules at each pre-trained layer, it significantly decreases the efficiency due to the number of task-specific parameters used during training and inference. To address this issue, we propose a pruning strategy to reduce the number of experts.

In order to decide which expert to keep at each layer, we first fine-tune the full set of expert modules using our Mixture-of-Linguistic-Experts architecture (Figure 1) for a fixed small number of steps. After the gates have converged, importance score from the gates can be used to determine which experts to keep. While an iterative pruning strategy (Michel et al., 2019; Behnke and Heafield, 2020; Tan and Motani, 2020) can also be used, it is less efficient due to the requirement of more training steps. Finally, after the pruning process, we restart the training process and fine-tune the resulting model with one expert module per layer.

## 5 Experiments

We describe in detail the experimental settings and results in this section. We start by providing a brief summary of the linguistic graphs (§5.1) before describing the datasets (§5.2) models (§5.3), and the hyperparameters settings (§5.4). Finally, we present the results in §5.5.

### 5.1 Linguistic Graphs

In our experiments, we use three different linguistic graphs to encode sentence-level structures. Following prior studies (Wu et al., 2021; Yu et al., 2022), we infuse the semantic and syntactic dependency trees as well as a sequential bidirectional graph into three separate RGCN adapter modules for each layer of the pre-trained model. In addition, to account for scenarios where structures are either not needed or harmful, we also use a multi-layer perception (MLP) module to represent an edgeless graph, where no composition is performed.

**Syntactic Trees** In syntactic parses, each word in the sentence is assigned a syntactic head based on Universal Dependencies (UD) formalism (de Marneffe et al., 2021). We use the Bi-LSTM-based deep biaffine neural dependency parser (Dozat and Manning, 2017) trained on the English UD treebank from the Stanza library (Qi et al., 2020).

**Semantic Trees** Based on the DELPH-IN dependencies formalism (Ivanova et al., 2012), semantic parses assign word dependencies based on predicate-argument relations. In contrast to syntactic graphs, words that do not contribute to the meaning representation of the sentence do not appear in the semantic graph. The graphs are extracted with a neural transition-based parser (Wang et al., 2018; Che et al., 2019) trained on the CoNLL 2019 shared task (Oepen et al., 2019).

**Sequential Bidirectional Graphs** We also use a straight-forward sequential bidirectional graph that connects word tokens in a sequential order. This allows the RGCN layers to aggregate local information rather than potentially long dependencies, where it has shown the ability to improve task performance when injected into pre-trained transformer layers via fixed attention (Li et al., 2022).

**Edgeless Graphs** In addition to the three linguistic graphs, we also apply a straight-forward nonlinear transformation using MLP layers. The intuition is that at some layers, injecting structures might be unhelpful (or even detrimental) to the task performance when the linguistic prior cannot be utilized based on the representation learned by that layer.

### 5.2 Datasets

| Dataset | Task | Train | Dev |
|---|---|---|---|
| CoLA | Acceptability | 1K | 1.74 |
| RTE | Entailment | 2.5K | 278 |
| MRPC | Paraphrase | 2.7K | 409 |
| STS-B | Similarity | 5.8K | 1.5k |
| SST-2 | Sentiment | 67K | 873 |
| QNLI | Entailment | 105k | 5.5K |
| QQP | Entailment | 363K | 40K |
| MNLI | Entailment | 392k | 9.8K |

Table 1: The statistics of the datasets in the GLUE benchmark, ordered by the size of the training set.

We conduct all our experiments on the GLUE benchmark (Wang et al., 2019a), consisting of a comprehensive suite of natural language understanding tasks. The benchmark contains eight datasets for text classification, including linguistic acceptability (CoLA), sentiment analysis (SST-2), similarity and paraphrase tasks (MRPC, STS-B, QQP), and natural language inference (MNLI, QNLI, RTE). For evaluation metric, we use Matthew's Correlation for CoLA, F1 for MRPC and QQP, Spearman's Rank-Order Correlation for STS-B, and Accuracy for SST-2, RTE, QNLI, and MNLI. Following prior studies (Houlsby et al., 2019b; He et al., 2022), we exclude the WNLI dataset from our experiments due to its limited coverage. The statistics of the datasets are presented in Table 1.

### 5.3 Models

In our experiments, we apply our methods to three different pre-trained language models: BERT, RoBERTa, DeBERTaV3. RoBERTa (Liu et al., 2019) enhances BERT (Devlin et al., 2019) by incorporating more training data and removing the next-sequence prediction objective, DeBERTa (He et al., 2021) introduced a disentangled attention mechanism for encoding relative positions at every layer, while DeBERTaV3 (He et al., 2021) improved upon the prior versions by adapting the

| Method | CoLA | RTE | MRPC | STS-B | SST-2 | QNLI | QQP | MNLI | Average |
|---|---|---|---|---|---|---|---|---|---|
| BERT | | | | | | | | | |
| Full Fine-Tuning | 62.08 | 66.43 | 90.94 | 89.76 | 91.63 | 89.95 | 87.35 | 83.23 | 82.67 |
| Adapter | 61.51 | 71.84 | 89.86 | 88.63 | 91.86 | 90.55 | 86.78 | 83.14 | 83.02 |
| Prefix-tuning | 55.37 | 76.90 | 91.29 | 87.19 | 90.94 | 90.39 | 83.30 | 81.15 | 82.07 |
| LoRA | 60.47 | 71.48 | 90.03 | 85.65 | 91.51 | 89.93 | 85.98 | 82.51 | 82.20 |
| UniPELT (AP) | 61.15 | 71.84 | 90.28 | 88.86 | 91.86 | 90.77 | 86.74 | 83.41 | 83.12 |
| UniPELT (APL) | 61.53 | 73.65 | 90.94 | 88.93 | 91.51 | 90.50 | 87.12 | 83.89 | 83.51 |
| Ours | 61.49 | 70.36 | 90.43 | 88.71 | **92.66** | **93.03** | **87.82** | **84.30** | 83.60 |
| RoBERTa | | | | | | | | | |
| Full Fine-Tuning | 68.0 | 86.6 | 90.9 | 92.4 | 96.4 | 94.7 | 92.2 | 90.2 | 88.9 |
| UniPELT (APL) | 61.91 | 74.31 | 91.74 | 90.26 | 93.92 | 92.00 | 87.68 | 87.23 | 84.88 |
| Ours | 62.20 | 72.32 | 92.77 | 90.34 | **94.27** | **92.53** | **88.29** | **87.83** | 85.07 |
| DeBERTaV3 | | | | | | | | | |
| Full Fine-Tuning | - | - | - | - | - | - | 90.7 | - | - |
| UniPELT (APL) | 68.02 | 81.59 | 92.42 | 91.70 | 95.64 | 93.65 | 89.60 | 89.13 | 87.72 |
| Ours | 69.93 | 79.42 | 93.38 | 91.01 | 95.84 | 93.92 | **89.83** | **89.47** | 87.85 |

Table 2: Results on the GLUE benchmark for BERT, RoBERTa, and DeBERTaV3. For BERT, the full fine-tuning and PEFT baseline results are directly copied from Mao et al. (2022). For both RoBERTa and DeBERTaV3, the UniPELT results are obtained using the AdapterHub implementation (Pfeiffer et al., 2020), while the full fine-tuning results are copied from their original papers (RoBERTa only reported precision to the tenth decimal place, while DeBERTaV3 only reported the base model on QQP). All our results are averages over three seeds, with statistically significant improvements (>99% confidence Bootstrap Test) over UniPELT highlighted in **bold**.

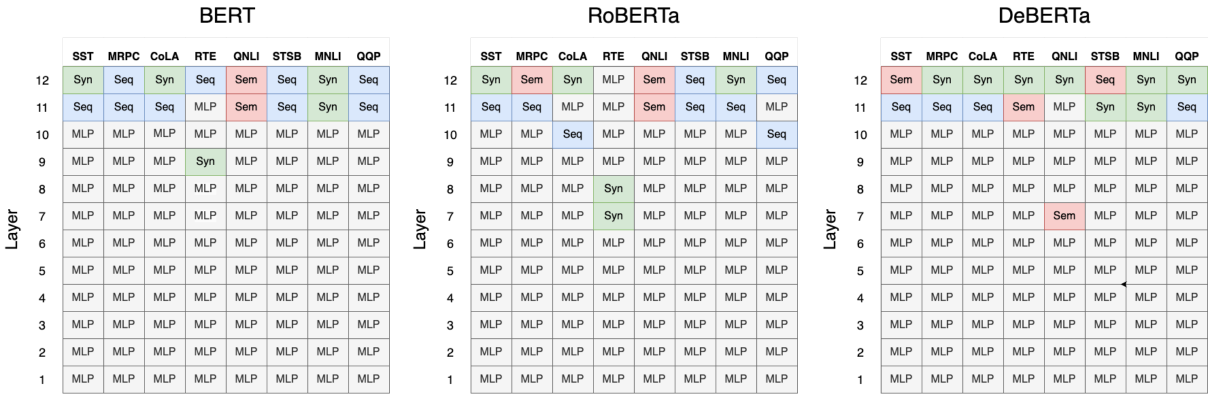
replaced token detection objective (Clark et al., 2020). For all models, we use the standard variant with 12 layers and 12 heads. For baselines, we use the unified framework for parameter-efficient language model tuning (UniPELT) proposed by (Mao et al., 2022). Since the results from the original paper demonstrated superior performance over other PEFT methods (Houlsby et al., 2019a; Li and Liang, 2021; Hu et al., 2022), we only report the results for these methods for BERT. For all tasks, we apply a classifier on the [CLS] token representation from the last hidden layer.

### 5.4 Hyperparameters

Both MLP and RGCN adapter modules consist of two hidden layers with a bottleneck dimension of 48. Since RGCN modules require 3× the number of parameters as MLP modules, we only select the top-2 RGCN modules based on their gate values. Following the settings by Mao et al. (2022), we set the input length to 128, and train for a total of 50 epochs for with a learning rate of $5e-4$ and batch size of 16. During the initial steps of training our Mixture-of-Linguistic-Experts model, we follow the suggestions from prior work (Huijben et al., 2022) and apply temperature annealing (Jang et al., 2017) to gradually decrease the temperature from $5$ to $0.1$ over $1000$ steps. The intuition behind temperature annealing is to allow the model to start with a more exploratory behavior before gradually becoming more exploitative. Lastly, we also scale the adapter output by a constant factor of $4$ as proposed in the work by He et al. (2022).

### 5.5 Results

From the results in Table 2, we see that our approach achieves the best overall performance on the GLUE benchmark. For individual tasks, although our method lags behind UniPELT in the low-resource tasks of RTE and STS-B, our method achieves consistent improvements in the four tasks with the highest number of training examples (Table 1): SST-2, QNLI, QQP and MNLI, where the improvements for SST-2 and QNLI are statistically significant for two out of the three models, and QQP and MNLI for all three models. This is consistent with the findings by prior work (Mao et al., 2022; Chen et al., 2022), where they found that while existing PEFT methods excel in low-resource tasks, they still struggle to yield consistently competitive performance in medium and high-resource settings. However, we believe that learning to use the linguistic structures associated with the dependency trees requires more tuning and can outper-



BERT

| Layer | SST | MRPC | CoLA | RTE | QNLI | STSB | MNLI | QQP |
|---|---|---|---|---|---|---|---|---|
| 12 | Syn | Seq | Syn | Seq | Sem | Seq | Syn | Seq |
| 11 | Seq | Seq | Seq | MLP | Sem | Seq | Syn | Seq |
| 10 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 9 | MLP | MLP | MLP | Syn | MLP | MLP | MLP | MLP |
| 8 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 7 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 6 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 5 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 4 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 3 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 2 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 1 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |

RoBERTa

| Layer | SST | MRPC | CoLA | RTE | QNLI | STSB | MNLI | QQP |
|---|---|---|---|---|---|---|---|---|
| 12 | Syn | Sem | Syn | MLP | Sem | Seq | Syn | Seq |
| 11 | Seq | Seq | MLP | MLP | Sem | Seq | Seq | MLP |
| 10 | MLP | MLP | Seq | MLP | MLP | MLP | MLP | Seq |
| 9 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 8 | MLP | MLP | MLP | Syn | MLP | MLP | MLP | MLP |
| 7 | MLP | MLP | MLP | Syn | MLP | MLP | MLP | MLP |
| 6 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 5 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 4 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 3 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 2 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 1 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |

DeBERTa

| Layer | SST | MRPC | CoLA | RTE | QNLI | STSB | MNLI | QQP |
|---|---|---|---|---|---|---|---|---|
| 12 | Sem | Syn | Syn | Syn | Syn | Seq | Syn | Syn |
| 11 | Seq | Seq | Seq | Sem | MLP | Syn | Syn | Seq |
| 10 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 9 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 8 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 7 | MLP | MLP | MLP | MLP | Sem | MLP | MLP | MLP |
| 6 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 5 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 4 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 3 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 2 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |
| 1 | MLP | MLP | MLP | MLP | MLP | MLP | MLP | MLP |

Figure 2: These heatmaps illustrate which of the four expert modules are used at each layer for the three models. We use green, red, blue, and grey to represent the syntactic (Syn), semantic (Sem), sequential (Seq), and MLP expert, respectively.

form standard PEFT methods when there are more training data available. Lastly, it is worth highlighting that the application of our approach to the RoBERTa model resulted in a notable increase in performance (+0.19) over the baseline, surpassing the gains observed with BERT (+0.09) and DeBERTa (+0.13). Since RoBERTa is pre-trained on a larger corpus than BERT, we hypothesize that this discrepancy could be due to the fact that RoBERTa has learned more meaningful representation for understanding linguistic structures. Conversely, the advantage of the injected linguistic structures could be somewhat offset by the more sophisticated pre-training methodology employed by DeBERTa. Lastly, we note that while Mao et al. (2022) reported that their method (UniPELT) achieved significantly better performance compared to standard fine-tuning[1], our experiments with RoBERTa yielded the opposite conclusion[2]. This is consistent with the findings by Chen et al. (2022), where they find that PELT methods are highly unstable and cannot achieve consistently competitive performance compared to fine-tuning (especially in medium- to high-resource settings). Therefore, we hypothesize the discrepancy between our results and theirs is due to the likely extensive hyperparameter search conducted by (Mao et al., 2022), whereas we used the identical hyperparameter settings across all experiments as reported in subsection 5.4.

In Table 3, we report the number of trainable parameters for each of the methods in Table 2. While increasing the number of RGCN modules or hidden layer dimensions could improve the performance of our model, our hyperparameters settings (subsection 5.4) are selected specifically to match the number of parameters used in UniPELT (Mao et al., 2022). Additionally, it is worth mentioning that we elected not to incorporate the dependency relations since the number of parameters in RGCN layers increases linearly with the number of relation types.

| Method | Parameters |
|---|---|
| Fine-tuning | 110M (100%) |
| Adapter | 895K (0.81%) |
| Prefix-tuning | 184K (0.17%) |
| LoRA | 295K (0.27%) |
| UniPELT (AP) | 1.1M (0.99%) |
| UniPELT (APL) | 1.4M (1.26%) |
| Ours | 1.2M (1.14%) |

Table 3: Number of trainable parameters required for each parameter-efficient fine-tuning method.

## 6 Analysis

In this section, we provide an analysis of the model's behavior by first examining the linguistic expert used for each model (§6.1) before examining the convergence rate of Gumbel-Softmax gates at different layers of the model (§6.2).

### 6.1 Gate Values

Figure 2 illustrates the experts used at each layer of the models. At first glance, we can clearly see that all models tend to favor RGCN modules at the upper layers, while the standard MLP adapter is used for lower layers. This could be due to

[1] Since the original BERT paper (Devlin et al., 2019) does not report the GLUE development set results, the full fine-tuning results for BERT in Table 2 are copied from Mao et al. (2022).

[2] The full fine-tuning results for RoBERTa and DeBERTaV3 are copied for their original papers (Liu et al., 2019; He et al., 2021), where He et al. (2021) only reported the full set of GLUE results for their large variant. In both papers, the reported results are limited to three significant digits.

the fact that pre-trained language models are designed to learn hierarchical representations of the input, where lower-level layers typically capture the surface knowledge required to understand high-order compositions (Tenney et al., 2019a; Niu et al., 2022). Since such knowledge are generally applicable to all downstream tasks with minimal modification even during full fine-tuning (Zhou and Srikumar, 2022), augmenting their representations with compositional structures could be detrimental to the performance. Similar findings have also been reported by Rücklé et al. (2021), where dropping out adapters in the lower layers has the least amount of impact on model performance.

To gain a better understanding of how different linguistic structures are utilized, we provide a qualitative comparison of linguistic experts used between models. From Figure 2, we can see that the experts used between BERT and RoBERTa are very similar, with 5 out of the 8 tasks being exactly the same. In contrast, DeBERTa tends to use more semantic and syntactic experts, with no sequential experts selected on the top layer. We believe this is due to the disentangled attention mechanism used by the DeBERTa model (He et al., 2021), where the token positions are already encoded by an additional vector at each layer of the model. Additionally, we see that semantic graphs are selected the least. This could be due to the fact that we do not model the relation types, which are necessary to determine the nuanced semantic roles between concepts and ideas. Conversely, the relation types in syntactic trees (e.g., subject-verb) do not provide the full meaning of the sentence beyond grammatical structure, where prior studies have shown that syntax trees with no relations can still be beneficial for downstream performance (Bai et al., 2021).

## 6.2 Gate Convergence

Next, we examine the convergence rate for the gate logits by measuring the changes in the gate value between steps. For the purpose of analysis, we train the full set of experts for 2000 steps while keeping all hyperparameters the same. Figure 3 plots the JS-Divergence between the gate values' softmax distribution in 10-step intervals. From the plot, we can see that the gate values in the lower layers change rapidly in the early iterations before converging. This implies that the model can quickly learn to select the MLP module (§6.1), providing further evidence against injecting structural knowledge at lower layers of pre-trained models. In the upper layers, while it follows a similar trend where the gates change quickly before the change curve flattens out, we still see a moderate amount of oscillation even after 1000 steps. This can be interpreted as the best expert not having enough relative advantage over the others for the model to assign a high importance score. Since the main purpose of our work is to propose an architecture for selecting experts, we leave the in-depth investigation regarding the trade-off between different linguistic experts as an interesting venue for future work. Finally,

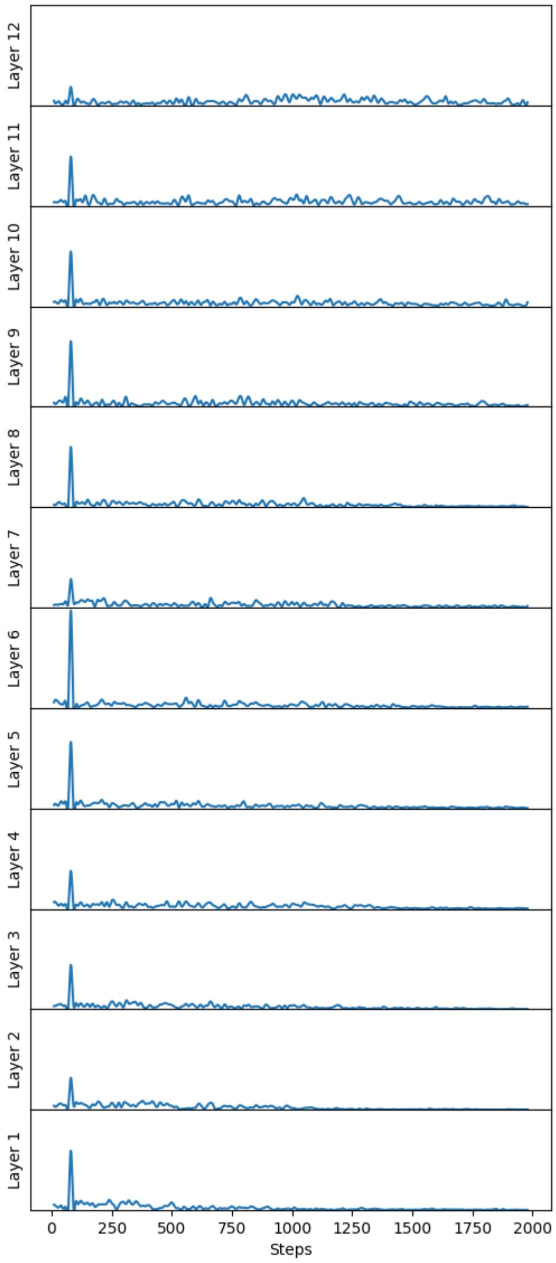


Figure 3: Average JS-Divergence between the gate value distribution, measured in 10-step intervals.

we see that almost all gates have converged at the 250-step mark. For reference, this is roughly 2% of the number of steps for a single epoch on the MNLI training set. This finding demonstrates that only a small number of steps are required to learn the importance of different adapter modules.

### 6.3 Ablation Study

We perform ablation experiments to study the effectiveness of our expert selection method based on the importance scores (section 4). To ensure a fair comparison with the results in Table 2, we only use one expert per layer while using the same architecture. We manually design the ordering of experts based on the intuition of the traditional NLP pipeline (Tenney et al., 2019a), with surface features at the bottom, syntactic features in the middle, and semantic features at the top (Jawahar et al., 2019). Specifically, we use sequential-graph encoding position information at the lower four layers, syntactic trees at the middle four layers, and semantic graph at the upper four layers. We also perform experience using only one expert for the entire model as a baseline.

| Method | SST-2 | QNLI |
|---|---|---|
| Syntax-only | 92.04 | 86.40 |
| Semantic-Only | 85.36 | 85.36 |
| Positional-Only | 88.97 | 88.97 |
| Manually-Designed | 91.84 | 89.12 |
| Ours | **94.27** | **92.53** |

Table 4: Ablation results with RoBERTa with manually selected experts, averaged across 3 seeds.

Table 4 shows the results for RoBERTa on the two medium-resource datasets (SST-2 and QNLI). From the results, we see that while our manually-designed approach achieved better performance than the single-expert models, they still significantly trail behind our automatic selection approach. This finding verifies our hypothesis that augmenting the representations of lower layers with compositional structures can have unrecoverable effects on the upper-layer representations used for task prediction (subsection 6.1), ultimately leading to a significant deterioration in performance.

## 7 Conclusion and Future Work

In this work, we introduce an approach that combines the two popular research areas of injecting linguistic structures and parameter-efficient fine-tuning (PEFT). To start, we introduce a novel framework that combines multiple linguistic structures in an architecture inspired by the Mixture-of-Experts model, where Gumbel-Softmax gates are used to learn the importance of these experts at different layers of the model in a small fixed number of training steps. Finally, we reduce the parameter count by pruning all but one expert at each layer such that the resulting number of trainable parameters is comparable to state-of-the-art PEFT methods. After running experiments with three different pre-trained models on the GLUE benchmark, the results show that our method can achieve the best overall performance while significantly outperforming the baselines on high-resource tasks. Finally, we examine the experts selected by each model and the convergence rate of the Gumbel-Softmax gates to gain a better understanding of the models' behavior and provide valuable insights for future studies on knowledge injection.

For future work, we plan to perform further experiments to determine the relative advantage of different linguistic knowledge and study how the quality of graphs affects model performance on downstream tasks. One significant challenge is to efficiently incorporate relation types of dependency trees, which we will explore in future work. In addition, we plan to further improve the efficiency of our approach by incorporating findings from other recent works, such as dropping adapters in the lower layers (Rücklé et al., 2021). Lastly, we plan to extend our approach to inject linguistic structures (including discourse graphs) into decoder-only architectures (Radford et al., 2019; Brown et al., 2020b) and perform studies on larger model variants (Touvron et al., 2023).

## Limitations

One limitation of our study is that our approach (excluding sequential graphs) requires high-quality parsers to construct gold-standard syntactic and semantic trees. While our approach is generally applicable to all structures, our experiments focus on sentence-level linguistic graphs on the GLUE benchmark. Other structures such as discourse trees on multi-sentential tasks remain to be explored in future studies. Additionally, all our experiments are performed on standard variants of pre-trained encoder models, different behavior could be observed on larger or differently structured models, such as the decoder-only architecture.

## References

David Arps, Younes Samih, Laura Kallmeyer, and Hassan Sajjad. 2022. Probing for constituency structure in neural language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6738–6757, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-BERT: Improving pre-trained transformers with syntax trees. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3011–3020, Online. Association for Computational Linguistics.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics.

Maximiliana Behnke and Kenneth Heafield. 2020. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2664–2674, Online. Association for Computational Linguistics.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Wanxiang Che, Longxu Dou, Yang Xu, Yuxuan Wang, Yijia Liu, and Ting Liu. 2019. HIT-SCIR at MRP 2019: A unified pipeline for meaning representation parsing via efficient training and effective encoding. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 76–85, Hong Kong. Association for Computational Linguistics.

Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022. Revisiting parameter-efficient tuning: Are we really there yet? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2612–2626, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

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.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

C. Goller and A. Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pages 347–352. IEEE.

Wenjuan Han, Bo Pang, and Ying Nian Wu. 2021. Robust transfer learning with pretrained language models through adapters. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 854–861, Online. Association for Computational Linguistics.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. 2019. Multi-granularity self-attention for neural machine translation. 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 887–897, Hong Kong, China. Association for Computational Linguistics.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*.

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.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019a. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019b. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

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*.

Binxuan Huang and Kathleen Carley. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. 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 5469–5477, Hong Kong, China. Association for Computational Linguistics.

Iris AM Huijben, Wouter Kool, Max B Paulus, and Ruud JG Van Sloun. 2022. A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1353–1371.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom? a contrastive study of syntacto-semantic dependencies. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 2–11, Jeju, Republic of Korea. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 565–576, Online. Association for Computational Linguistics.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. 2017. What do recurrent neural network grammars learn about syntax? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1249–1258, Valencia, Spain. Association for Computational Linguistics.

Adhiguna Kuncoro, Lingpeng Kong, Daniel Fried, Dani Yogatama, Laura Rimell, Chris Dyer, and Phil Blunsom. 2020. Syntactic structure distillation pretraining for bidirectional encoders. *Transactions of the Association for Computational Linguistics*, 8:776–794.

Jaejun Lee, Raphael Tang, and Jimmy Lin. 2019. What would elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Raymond Li, Wen Xiao, Linzi Xing, Lanjun Wang, Gabriel Murray, and Giuseppe Carenini. 2022. Human guided exploitation of interpretable attention patterns in summarization and topic segmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10189–10204, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. 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. *arXiv preprint arXiv:1907.11692*.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Scott Yih, and Madian Khabsa. 2022. UniPELT: A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6253–6264, Dublin, Ireland. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online. Association for Computational Linguistics.

William Merrill, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. 2021. Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? *Transactions of the Association for Computational Linguistics*, 9:1047–1060.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv preprint arXiv:2111.01243*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. 2021. Refining targeted syntactic evaluation of language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3710–3723, Online. Association for Computational Linguistics.

Jingcheng Niu, Wenjie Lu, and Gerald Penn. 2022. Does BERT rediscover a classical NLP pipeline? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3143–3153, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Stephan Oepen, Omri Abend, Jan Hajic, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdenka Uresova. 2019. MRP 2019: Cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 1–27, Hong Kong. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Jakob Prange, Nathan Schneider, and Lingpeng Kong. 2022. Linguistic frameworks go toe-to-toe at neuro-symbolic language modeling. In *Proceedings of the*

*2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4375–4391, Seattle, United States. Association for Computational Linguistics.

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

Peng Qian, Tahira Naseem, Roger Levy, and Ramón Fernandez Astudillo. 2021. Structural guidance for transformer language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3735–3745, Online. Association for Computational Linguistics.

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

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1192–1202, Berlin, Germany. Association for Computational Linguistics.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. 2021. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online. Association for Computational Linguistics.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 593–607. Springer.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea. Association for Computational Linguistics.

Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng, and Christopher D. Manning. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, page 129–136.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Chong Min John Tan and Mehul Motani. 2020. Dropnet: Reducing neural network complexity via iterative pruning. In *International Conference on Machine Learning*, pages 9356–9366. PMLR.

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*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Sean Trott, Tiago Timponi Torrent, Nancy Chang, and Nathan Schneider. 2020. (Re)construing meaning in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5170–5184, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online. Association for Computational Linguistics.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019b. Self-attention with structural position representations. 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 1403–1409, Hong Kong, China. Association for Computational Linguistics.

Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. 2019c. Tree transformer: Integrating tree structures into self-attention. 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 1061–1070, Hong Kong, China. Association for Computational Linguistics.

Yuxuan Wang, Wanxiang Che, Jiang Guo, and Ting Liu. 2018. A neural transition-based approach for semantic dependency graph parsing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Wei Wu, Houfeng Wang, Tianyu Liu, and Shuming Ma. 2018. Phrase-level self-attention networks for universal sentence encoding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3729–3738, Brussels, Belgium. Association for Computational Linguistics.

Zhaofeng Wu, Hao Peng, and Noah A. Smith. 2021. Infusing finetuning with semantic dependencies. *Transactions of the Association for Computational Linguistics*, 9:226–242.

Changlong Yu, Tianyi Xiao, Lingpeng Kong, Yangqiu Song, and Wilfred Ng. 2022. An empirical revisiting of linguistic knowledge fusion in language understanding tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10064–10070, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9628–9635.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Yichu Zhou and Vivek Srikumar. 2022. A closer look at how fine-tuning changes BERT. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1046–1061, Dublin, Ireland. Association for Computational Linguistics.