PALs and MNRL: Adaptations for Multi-Task BERT

Stanford CS224N Default Project

Lei YIN
Department of Computer Science
Stanford University
yinlei@stanford.edu

Abstract

This study explores advanced fine-tuning techniques for the BERT model to enhance its performance on three sentence-level tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. We focus on refining BERT's embeddings for efficient multi-task learning while maintaining individual task accuracy. We evaluate new architecture known as Projected Attention Layer (PAL), different Loss Functions and Optimization such as Multiple Negatives Ranking Loss (MNRL), Curriculum learning and oversampling, and various pooling strategies, proposing a novel hybrid multi-task training curriculum that surpasses traditional approaches. Notably, we achieve an overall accuracy of 0.653 on the test set, with task-specific improvements to 0.486 (SST), 0.796 (Quora), and 0.404 (STS), demonstrating advancements in multi-task learning with BERT.

1 Key Information to include

- Custom or Default Project: Default
- TA Mentor: Soumya Chatterjee
- External Collaborators (if you have any): No
- Sharing project: No

2 Introduction

Natural Language Processing (NLP) stands at the forefront of enabling machines to understand, interpret, and respond to human language in a valuable way. Among the plethora of tasks that NLP tackles, sentence-level undertakings such as sentiment analysis, paraphrase detection, and semantic textual similarity are pivotal for a range of applications, from automated customer service to content analysis and beyond. However, the intricacies of human language, including its implicit meanings, nuances, and context dependencies, make these tasks particularly challenging.

The advent of models like BERT Bidirectional Encoder Representations from Transformers Delvin et al. (2018) has significantly advanced the field's capabilities. However, while BERT provides a robust foundation for addressing diverse NLP tasks, its standard fine-tuning process often leads to suboptimal performance when applied across multiple distinct tasks simultaneously. This limitation primarily arises due to task interference, where the learning from one task can negatively affect the performance on another, and the challenge of effectively balancing multiple objectives within a single model architecture.

Current methods attempt to mitigate these issues through various strategies, such as task-specific layers or fine-tuning separate models for each task. However, these approaches can be inefficient and fail to exploit the potential synergies between tasks. Additionally, they may not adequately address the problem of loss imbalance, where differences in task difficulty or dataset size lead to skewed learning priorities.

This study aims to address these challenges by exploring advanced fine-tuning methodologies that leverage the strengths of the BERT model while minimizing the weaknesses associated with multi-task learning. By incorporating Projected Attention Layers (PALs), developed by Stickland and Murray (2019), loss-balancing techniques, Multiple Negatives Ranking Loss Learning Henderson et al. (2017), and an originally developed multi-task training curriculum, this study seeks to harmonize the learning objectives across different tasks, thereby reducing task interference and promoting effective knowledge transfer.

Our results demonstrate notable improvements over baseline models, achieving an overall accuracy of 0.653 on a composite test dataset. Specifically, we observed scores of 0.503 on the SST dataset (sentiment analysis), 0.796 on the Quora dataset (paraphrase detection), and 0.404 on the STS dataset (semantic textual similarity). These outcomes not only signify the effectiveness of our approach in enhancing multi-task learning but also offer valuable insights into the dynamics of task interactions within the BERT framework.

In the following sections, we delve deeper into the problem domain, review related works, outline our methodology, present detailed experimental results, and discuss the implications of our findings for future research in multi-task NLP.

3 Related Work

Parameter-efficient Strategies The technique of fine-tuning pre-trained models has emerged as a powerful transfer learning approach, notably through the success of BERT Delvin et al. (2018). While BERT's architecture has provided a solid foundation for addressing diverse NLP challenges, the process of fine-tuning presents a challenge in terms of parameter efficiency, particularly when adapting to multiple downstream tasks. This limitation forms the basis for our exploration into more parameter-efficient strategies.

Stickland and Murray (2019) introduced the concept of Projected Attention Layers (PALs), providing an innovative method to boost BERT's multi-task learning capabilities by integrating small, task-specific layers. This approach helps maintain the model's parameter efficiency while enhancing its adaptability to various tasks. In this project, we have implemented three types of methodologies from these studies and have combined two of them to enhance our baseline method, aiming to effectively optimize multi-task performance.

Multiple Negative Ranking Loss Henderson et al. (2017) is a powerful technique for learning effective representations in tasks involving sentence pairs, such as paraphrase detection and semantic textual similarity. By considering multiple negative examples during training, MNRL helps the model learn to distinguish between similar and dissimilar sentence pairs more effectively. It encourages the model to push away the representations of negative examples while pulling the representations of positive pairs closer together. This contrastive learning approach enables the model to capture fine-grained semantic differences and similarities between sentences. MNRL has been shown to improve performance on various sentence pair tasks and enhance the quality of the learned sentence embeddings. In this project we explored MNRL implementation and the performance of different combinations with cross-entropy loss.

Loss-balance Strategies The loss imbalance problem in multitask learning refers to the challenge of balancing different tasks during the training process, where differences in the importance, scale, or difficulty of tasks can lead to suboptimal learning and performance. In multitasking settings, it's crucial to ensure that no single task dominates the training process, allowing all tasks to contribute effectively to the learning of shared features.

SMOTE (Synthetic Minority Over-sampling Technique) Chawla et al. (2002) is a technique used to address class imbalance in datasets by generating synthetic samples for the minority class. By interpolating between existing minority class examples, SMOTE creates new, similar instances, which helps to balance the class distribution and improve the performance of classifiers on imbalanced datasets. Curriculum Learning Bengio et al. (2009) is a training strategy inspired by the way humans learn, which involves starting with simpler tasks and gradually introducing more complex ones. By initially focusing on easier examples and progressively increasing the difficulty, curriculum learning helps models to build a solid foundation and achieve better generalization and performance. In this

study, we use these methods as an extension of the baseline method to simultaneously improve the multi-task performance.

4 Approach

Baseline
The backbone of baseline is identical to part one and part two of the default project. Perform sequential training on each dataset, and adopt default CLS pooling strategy mentioned in original BERT paper (Delvin et al. (2018)). I used the combined loss function, the simple sum of the loss functions for each task:

$$\text{Loss} = \text{Loss}_{\text{sentiment}} + \text{Loss}_{\text{paraphrase}} + \text{Loss}_{\text{similarity}}$$

Projected Attention Layer and other Parameter Adaptions
Firstly, the BERT layer can be expressed as:

$$BL(h) = LN(LN(h + MH(h)) + FFN(LN(h + MH(h))))$$

In the notation, BL stands for a BERT layer, LN represents layer normalization, FFN denotes the standard feed-forward network with d_{ff} hidden states, and MH refers to the multi-head attention layer, where h is the hidden vector with dimension d. Consequently, a BERT layer comprises $4d^2 + 2dd_{ff}$ parameters. Setting $d = d_m$ and $d_{ff} = 4d_m$, the BERT layer possesses $12d_m^2$ parameters.
The architecture of the first method, known as the Projected Attention Layer (PAL)(Stickland and Murray (2019)), the formulation is as follows:

$$BL_{PAL}(h) = LN(LN(h + MH(h)) + FFN(LN(h + MH(h))) + TS(h))$$

Here, the hidden vector h from the previous layer is forwarded not only to the multi-head attention layer (MH) but also to a task-specific layer, denoted as TS. The TS for PAL is given by:

$$TS(h) = V^D(MH(V^E(h)))$$

where V^E is an encoder of size $d_m \times d_s$, with d_s being significantly smaller than d_m, thereby primarily serving to reduce dimensionality. Conversely, V^D acts as the inverse process to V^E, with dimensions $d_s \times d_m$, and serves as a decoder for expanding the reduced dimensions.
The second method, called the low-rank layer, possesses an architecture and formula nearly identical to that of the PAL. The sole distinction lies in its task-specific layer, TS, which is modified as follows:

$$TS(h) = V^D(I(V^E(h)))$$

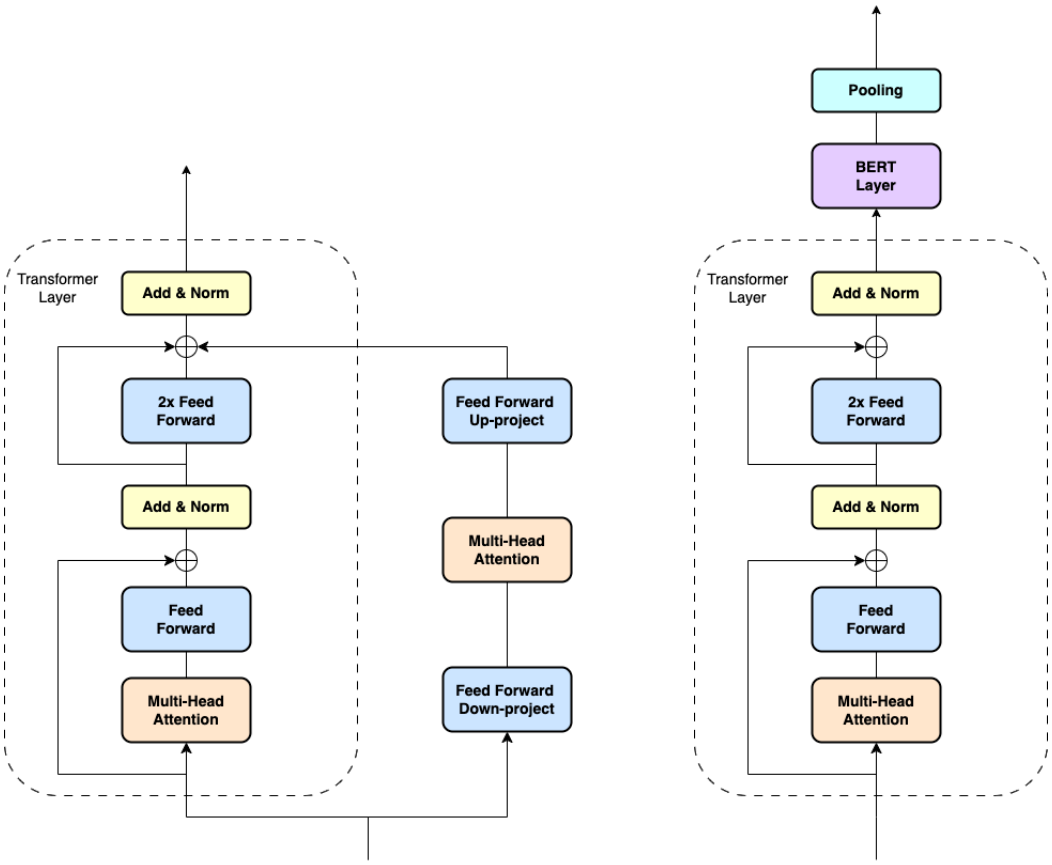
Here, the multi-head attention layer is replaced with an identity matrix, transforming this layer into a straightforward low-rank projection.
The third approach involves simply adding a task-specific BERT layer atop the shared BERT model. The architecture of PAL and Projected Attention on Top, is depicted in Figure 1.

The total parameters required for these methods are detailed in Table 1, where T represents the number of tasks, which totals three in this project. It is important to note that the PAL method shares V_E and V_D across different layers, but not across tasks.
Loss Function Optimization

METHOD	PARAMETERS
PAL	$T(2d_m d_s + 12 \times 3d_s^2)$
LOW RANK	$T(12 \times 2d_m d_s)$
PROJ. ATTEN. ON TOP	$T(12d_m^2)$

Table 1: Total parameters for various parameter adaption methods

Another effective way of improving embeddings would be to fine-tune the model with different loss functions. One approach would be Multiple Negative Ranking Loss(Henderson et al. (2017)). With this loss function, training data consists of sets of K sentence pairs $[(a_1, b_1), \ldots, (a_n, b_n)]$ where



(a) PAL (b) Projected Attention on Top

Figure 1: Architecture of PAL and parameter adaptions

a_i, b_i are labeled as similar sentences and all (a_i, b_j) where $i \neq j$ are not similar sentences. The loss function then minimizes the distance between a_i, b_i while it simultaneously maximizes the distance (a_i, b_j) where $i \neq j$. Specifically, training is to minimize the approximated mean negative log probability of the data. For a single batch, this is calculated as

$$\begin{aligned}\mathcal{J}(x, y, \theta) &= -\frac{1}{K}\sum_{i=1}^{K} \log P_{\text{approx}}(y_i \mid x_i) \\ &= -\frac{1}{K}\sum_{i=1}^{K}\left[S(x_i, y_i) - \log \sum_{j=1}^{K} e^{S(x_i, y_j)}\right]\end{aligned}$$

where θ represents the word embeddings and neural network parameters used to calculate S, a scoring function.

Pooling Strategies

Pooling is a critical operation used in the context of processing sequences, especially when adapting the model to tasks that require a fixed-size input vector, such as classification or similarity tasks. The three most commonly used pooling strategies are:

- CLS Token Pooling, which uses CLS embedding to represent the whole sentence.
- Mean Pooling, which takes the average of all token embeddings.
- Max Pooling, which takes the maximum value across all token embeddings for each dimension of the embedding space.

5 Experiments

5.1 Data

Sentiment analysis: SST dataset; Paraphrase detection: Quora dataset; Semantic textual similarity: SemEval.

5.2 Evaluation method

Sentiment classification:classification accuracy; Paragraph detection:binary prediction accuracy; Semantic textual similarity: Pearson correlation.

5.3 Experimental details

For the pre-trained BERT model, its dimension d_m is set to 768, and it consists of 12 layers. Regarding the parameter adaptation method, the dimension d_s is set to 256. Additionally, for training, models utilize a default learning rate of $1e-5$, a batch size of 32, and are trained for 10 epochs unless specifies otherwise.

5.4 Results

Firstly, I implemented various PAL architectures and experimented with different hyper-parameter settings. Initially, I used a basic PAL with a single linear layer for both the encoder and decoder, setting the dimension size to 128. However, this implementation did not yield significant improvements. Increasing the dimension size d_s to 256 resulted in enhanced performance across all tasks, including SST accuracy, Paraphrase accuracy, and STS correlation.

I also experimented with higher epochs, specifically up to 20, which further improved SST accuracy, Paraphrase accuracy, and STS correlation. Furthermore, I introduced separate encoder and decoder layers (defaulting to 2 layers) for each task (sentiment, paraphrase, and similarity). During the forward pass, the corresponding encoder and decoder layers are selected and applied based on the task. This modification significantly boosted Paraphrase accuracy and STS correlation, although it slightly reduced SST accuracy.

Additionally, I experimented with 3 projection layers to allow for more complex transformations of the hidden states. However, this increased the training time considerably and did not improve overall performance, indicating diminishing returns.

Next, I replaced the task-specific layer in the Projected Attention Layer (PAL) with a simple low-rank projection. Specifically, instead of passing the projected representation through the multi-head attention layer, it directly applied an identity mapping followed by a projection back to the original dimensionality.

Next, I combined the low-rank projections with a task-specific BERT layer on top. This combination yielded the highest overall performance scores.

PAL Architecture and Hyper-Parameter	SST	Paraphrase	STS	Overall
Baseline	0.502	0.730	0.365	0.638
Basic PAL, $d_s = 128$	0.499	0.720	0.382	0.637
Basic PAL, $d_s = 256$	**0.506**	0.731	0.380	0.639
Basic PAL, $d_s = 256$, epochs = 20	0.494	0.749	0.387	0.646
Multi-layer PALs, 2 projection layers	0.494	0.742	0.393	0.644
Multi-layer PALs, 3 projection layers	0.487	0.728	0.378	0.635
Low rank	0.487	0.728	0.378	0.635
Low rank + PALs on top	0.494	**0.744**	**0.394**	**0.647**

Table 2: Experiments on dev set with different PAL architecture

After conducting experiments on various PAL architectures, I compared different pooling strategies with the baseline CLS pooling.

Architecture and Hyper-Parameter	SST	Paraphrase	STS	Overall
Low rank + PALs on top, CLS pooling	0.494	**0.744**	**0.394**	**0.647**
Low rank + PALs on top, mean pooling	**0.508**	0.713	0.390	0.639
Low rank + PALs on top, max pooling	0.501	0.713	0.384	0.635

Table 3: Experiments on dev set with different pooling strategy

After experimenting with different pooling strategies, I implemented the Multiple Negative Ranking Loss (MNRL). I attempted to apply MNRL to all tasks and also explored combinations such as using Cross-Entropy Loss for the sentiment task and MNRL for the other two tasks. The performance of

Architecture and Hyper-Parameter	SST	Paraphrase	STS	Overall
Low rank + PALs on top, mean pooling	0.508	**0.713**	**0.390**	**0.639**
Plus MNRL for all tasks	0.126	0.632	0.119	
Cross-Entropy Loss for SST + MNRL	**0.513**	0.491	0.049	

Table 4: Experiments on dev set with MNRL

MNRL is significantly worse than expected. I hypothesize that this could be due to the following reasons:

- Negative Sampling Strategy: The quality and relevance of negative samples are crucial for MNRL. If the negative samples are not representative, the model might not learn effective discriminative features. In the current implementation, every sample in the batch is used as a negative for others. This could lead to a scenario where many negatives are too easy, thus not providing a strong learning signal.
- Loss Balance: The combination of Cross-Entropy Loss and MNRL might need to be balanced. If MNRL dominates the training, it might not allow the model to learn well for each task.

Finally I applied curriculum learning, combined together with oversampling to address the imbalance and potentially improve the performance of the sentiment and STS tasks. I oversampled sentiment and STS training dataset to max 5 times more than its original size, and then train the model first on the larger dataset (Quora) and then gradually introduced to the smaller datasets (SST and STS) Chen and He (2021), Chen et al. (2018). Interestingly it significantly boosts Paraphrase and overall performance, but dropped on smaller dataset SST and STS. It could be attributed to

- Imbalanced Training Focus: The training process might still be disproportionately favoring the larger dataset. Quora dataset has significantly more data, which can dominate the learning process.
- Overfitting on Small Datasets: Since the model sees the same data multiple times, it might not generalize well on the sentiment and STS tasks, resulting in poorer performance on these tasks.

Such implementation also significantly increased training time to more than 12hours with GPU, and GCP became really un-stable and frequently drop connection during the training. Given more time, I'd pursue further fine tune on curriculum learning for a more optimal result.

Architecture and Hyper-Parameter	SST	Paraphrase	STS	Overall
Low rank + PALs on top, mean pooling	0.508	0.713	**0.390**	0.639
Plus oversampling and curriculum learning	0.486	**0.795**	0.344	**0.651**

Table 5: Experiments on dev set with oversampling and curriculum learning

6 Analysis

We further analyzed matrices of evaluation performance on PAL architecture. For SST, The gap

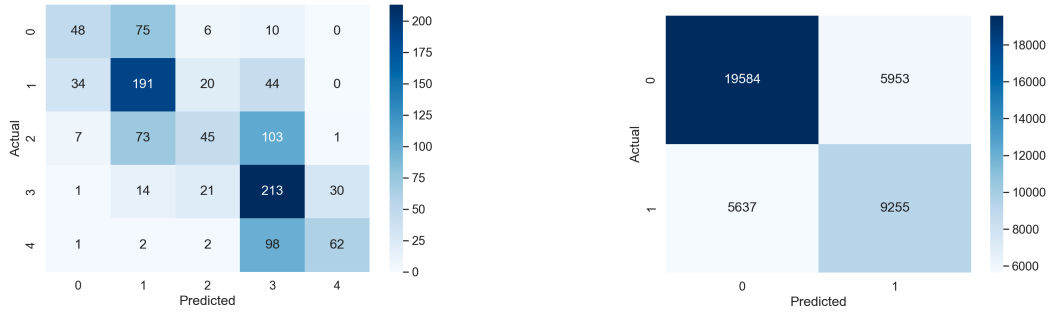
	SST	Paraphrase
Accuracy	0.508	0.713
F1 score	0.468	0.693
Precision	0.533	0.693
Recall	0.468	0.694

Table 6: Accuracy, precision, recall, and F1 scores

between precision (53.3%) and recall (46.8%) suggests that the model has a higher rate of false negatives than false positives, indicates that the model is somewhat biased towards predicting certain classes over others. The model may be more conservative in predicting positive instances, leading to lower recall. The F1 score being lower than both precision and accuracy suggests that the model's errors are not evenly distributed among classes. This is supported by the confusion matrix, which shows considerable misclassification among the middle categories.

For Paraphrase, the closeness of precision and recall in paraphrase detection suggests a balanced performance in terms of false positives and false negatives. This means the model is equally good at identifying actual paraphrases and correctly predicting non-paraphrases. When precision and recall are nearly equal, it indicates that the model has a good balance between sensitivity (recall) and specificity (precision). The high accuracy and closely aligned other metrics indicate that the model is performing consistently across different aspects of performance evaluation.

We also generated the confusion matrices for these two tasks. Below figure reveals that for SST the model rarely mispredicts label 4 as 0 or label 0 as 4, commonly mistaking neighboring labels (e.g., predicting 3 as 4 or 2 as 1). Concerning Paraphrase, it is also evident that a balanced performance in terms of false positives and false negatives.



(a) SST (b) Paraphrase

Figure 2: Confusion Matrices

7 Conclusion

In summary, this study involved exploring various modifications to enhance the performance of a multi-task BERT model. Adding multi-layer PALs proved to be an effective strategy. A comparison of seven parameter adaptation methods revealed that combining the low-rank and top attention method yielded the best PAL architecture. However, employing MNRL did not improve overall performance. Switching from CLS pooling to mean pooling did not show meaningful improvements either. The hybrid multi-task training curriculum together with oversampling was proposed, which outperformed both the sequential training approach and the naive multi-task training method. Furthermore, a combined solution of PAL architecture, curriculum learning and oversampling yielded the best result. Incorporating these findings, the developed multi-BERT model achieved scores of 0.653 on the development set and 0.651 on the test set, indicating that it does not suffer from overfitting. An analysis of the precision and recall scores identified a significant imbalance between the number of false positives and false negatives in the paraphrase detection task, providing direction for future work. In conclusion, this study provided an opportunity to explore various techniques for fine-tuning the multi-task BERT model and offered valuable practical experience in the field of natural language

processing. The insights gained from this study can be applied to further improve the performance of multi-task models and address specific challenges in tasks such as paraphrase detection.

8 Ethics Statement

Ethical challenges and societal risks associated with BERT-based NLP projects include:

- **1 Bias and Fairness**: BERT models can inherit biases present in the training data, leading to biased outputs or decisions. This can perpetuate or amplify societal biases and discrimination against certain groups. Mitigation strategies:
  - Carefully curate and diversify training data to reduce bias.
  - Implement bias detection and mitigation techniques during model training and evaluation.
- **2 Privacy and Data Protection**: Training BERT models requires large amounts of data, which may include sensitive or personal information. Improper handling or leakage of this data can violate individuals' privacy rights. Mitigation strategies:
  - Implement strict data protection measures and adhere to privacy regulations.
  - Anonymize or pseudonymize sensitive data before using it for training.
- **3 Misuse and Malicious Applications**: BERT-based models can be misused for malicious purposes, such as generating fake news, impersonation, or spreading disinformation. This can have serious societal consequences, undermining trust and manipulating public opinion. Mitigation strategies:
  - Implement safeguards to prevent misuse, such as watermarking generated content or restricting access to the model.
  - Educate users about responsible use and the potential risks of misuse.

It is crucial to proactively address these ethical challenges and risks throughout the development and deployment of BERT-based NLP projects. By implementing appropriate mitigation strategies, engaging in responsible practices, and fostering open dialogue, researchers and practitioners can work towards maximizing the benefits of these technologies while minimizing potential harm to individuals and society.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 321–357. AAAI Press.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758.

Zhao Chen, Vijay Badrinarayanan, Chen-yu Lee, and Andrew Rabinovich. 2018. Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR.

Jacob Delvin, Chang Ming-Wei, Lee Kenton, and Toutanova Kristina. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *arXiv preprint arXiv:1810.04805*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. In *arXiv preprint arXiv:1705.00652*.

Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In *International Conference on Machine Learning, pages 5986–5995. PMLR*.