Enhacing minBERT by Leveraging CosineEmbeddingLoss Fine-Tuning and Multi-Task Learning with Gradient Surgery

Stanford CS224N Default Project

Peter De La Cruz
Department of Computer Science
Stanford University
pdlcruz@stanford.edu

Mohamed Musa
Department of Computer Science
Stanford University
musam25@stanford.edu

Yahaya Ndutu
Department of Computer Science
Stanford University
yahayan@stanford.edu

Abstract

This project focuses on enhancing the semantic capabilities of a miniBERT model by integrating Multiple Negatives Ranking Loss and CosineEmbeddingLoss. Our initial efforts involved scaling down the BERT model and implementing essential features such as multihead self-attention and transformers. We aim to significantly improve performance in sentiment analysis and paraphrase detection by employing advanced techniques like Gradient Surgery for multi-task learning. Evaluation will utilize Spearman's rank correlation to measure semantic textual similarity and assess computational efficiency against established benchmarks like Sentence-BERT. Ultimately, we seek to elevate miniBERT's efficiency and accuracy, making it a viable tool for real-time NLP applications.

1 Key Information to include

- TA mentor: Jingwen Wu
- External collaborators: No
- External mentor: No
- Sharing project: No

2 Introduction

Efficient multi-task learning has been a long-term challenge to researchers in the world of deep learning, largely due to the interference of gradients in simultaneously learned tasks. This interference results in suboptimal performance and reduced data efficiency, especially when compared with models that are independently trained on tasks. This presents widespread implications, particularly in real-world applications where multi-task learning systems are prevalent, such as robotics, natural language processing, and computer vision. BERT (Devlin et al. 2018) is a current baseline model that leverages bidirectional pre-training and offers a wide range of tasks that avoids significant task-specific architecture modifications. For this project, we leverage a minimal version of BERT called miniBERT in order to improve sentiment analysis, paraphrase detection, and semantic textual similarity (STS). We investigate the integration of CosineEmbeddingLoss and Gradient Surgery for multi-task learning into the miniBERT model and its performance across various NLP tasks. This project ultimately

seeks to explore answers to the following question: Can the efficiency and accuracy of miniBERT be significantly improved by incorporating advanced techniques from research, namely Sentence-BERT's (Thakur et al., 2020) approach to embedding calculations and multi-task learning's Gradient Surgery approach? We namely incorporate the following enhancements: CosineEmbeddingLoss, Gradient Surgery, and Siamese and Triplet Network Structures.

3 Related Work

Multitask learning has been extensively explored within NLP. The exploration has been aimed to improve model generalizability by leveraging shared representations across related tasks. The aforementioned is evident in [1], where the attention mechanism introduced in transformers significantly impacts various NLP tasks, allowing for efficient processing and improved performance across multi-task learning scenarios. The transformer architecture's ability to handle multiple tasks simultaneously makes it an essential component in multitask learning frameworks.

Furthermore, in the Simple contrastive learning of sentence embeddings [2], a simple contrastive learning approach for sentence embeddings is presented, an approach that enhances the quality of embeddings by training with both supervised and unsupervised methods. This then shows an improved performance in semantic textual similarity tasks, which is critical for multitask learning where sentence embeddings are commonly used as shared representations.

Lora [3] makes another important contribution by introducing a low-rank adaptation technique for large language models, a technique which enables efficient model adaptation without significant computational overhead. This approach is particularly useful in multitask learning settings, where adapting models to new tasks while maintaining performance on existing tasks is crucial.

We also need to point out contributions that have aimed towards exploring challenges facing multitask learning. Overfitting and underfitting being some of the most common challenges especially when ealing with datasets of varying sizes, are explored in [4]. Here, we see Stickland, Asa Cooper, Murray, and Iain explore a unique training schema to gradually introduce all tasks during training, and balance the negative effects of different-sized datasets. This ensures more even training across tasks, particularly for those with less data. To complement the findings from above, one can also explore [5], where jiang et aal, implemented Smoothness Inducing Adversarial (SMART) regularization and Bregman Proximal Point Optimization (BPPO) to reduce overfitting and improve transfer learning. Using a combination of these two methods, they consistently outperformed the base BERT model across all 8 GLUE tasks by an immense margin, demonstrating improved model generalization in multitask domains.

[6] is also an important reference, as it explores Multiple Negatives Ranking loss (MNRL), designed to improve the model embeddings. This is done by mapping similar sentences closer together and dissimilar sentences farther apart. Building upon this, the ealrier mentioned approaches as well as other existing approaches our research aims to synthesize a model that provides the best accuracies leveraging the best of each extension.

4 Approach

4.1 Implementing miniBERT

Our baseline model is a streamlined miniBERT, as detailed in [7], which we have adapted to function with limited computational resources. This miniBERT implementation involves the essential functionalities such as the optimizer step, multihead self-attention, and transformers. We will successfully implemented these core features to operate the basic miniBERT model. Moving forward, we will focus on integrating Gradient Surgery, and CosineEmbeddingLoss to enhance its performance in sentiment analysis and paraphrase detection.

The methodological framework of this project incorporated the following enhancements to miniBERT:

- **CosineEmbeddingLoss Integration:** For optimizing the model during the training on tasks like semantic similarity [8].

- **Gradient Surgery:** Implementing the Gradient Surgery technique from multi-task learning to manage conflicting gradients, improving overall training efficiency and task performance[9].
- **Siamese and Triplet Network Structures:** Inspired by Sentence-BERT, these structures are used to generate embeddings that significantly reduce the time required for similarity calculations in semantic tasks[8].

4.2 Schedule Optimization

We optimized the training schedule using a method similar to round-robin scheduling to ensure efficient use of computational resources and balanced training across tasks.

- **Task Scheduling:** During each training epoch, batches are drawn in a round-robin manner from each task-specific dataset, ensuring that the model receives balanced training on all tasks.
- **Dynamic Batch Allocation:** The size of the batches for each task is dynamically adjusted based on the model's performance, ensuring that more challenging tasks receive more attention.

4.3 Cosine Embedding Loss

We optimized the training schedule using an approach similar to round-robin scheduling. This method ensures efficient utilization of computational resources and maintains balanced training across multiple tasks.

- Cosine-Similarity Embedding Loss: This loss function is defined as:

$$\text{CosineEmbeddingLoss}(x, y) = \begin{cases} 1 - \cos(x_1, x_2) & \text{if } y = 1 \\ \max(0, \cos(x_1, x_2) - \text{margin}) & \text{if } y = -1 \end{cases}$$

  where x_1 and x_2 are the embedding vectors, and y indicates whether the pair is similar or dissimilar. The margin parameter dynamically adjusts during training to optimize performance.

4.4 Gradient Surgery (PCGrad Rule)

To address the issue of gradient interference in multi-task learning, we implement the PCGrad (Projected Conflicting Gradients) rule.

- Key Theoretical Insights - Conflicting gradients are defined when the cosine similarity between them is negative:

$$\cos(\phi_{ij}) = \frac{\vec{g_i} \cdot \vec{g_j}}{\|\vec{g_i}\|\|\vec{g_j}\|} < 0$$

- The PCGrad algorithm is central to the implementation of the gradient surgery technique. This algorithm operates by adjusting the gradients of each task based on their interaction with gradients from other tasks within the same model, and will be explored more below.

PCGrad Algorithm: The algorithm modifies the gradient updates to mitigate the negative interference between conflicting tasks. For each task-specific gradient g_i, if it conflicts with another gradient g_j (i.e., their dot product is negative), g_i is projected onto the normal plane of g_j:

$$g_i \leftarrow g_i - \frac{g_i \cdot g_j}{\|g_j\|^2} g_j$$

This ensures that the conflicting components are removed, allowing for more effective and efficient multi-task learning.

The detailed PCGrad update rule is as follows:

Algorithm 1 PCGrad Update Rule

Require: Model parameters θ, task minibatch $B = \{T_k\}$
Compute gradients for each task $g_k = \nabla_\theta L_k(\theta)$ for all k
Initialize projected gradients $g_k^{\text{PC}} = g_k$ for each task
for each pair of tasks (i, j) **do**
    if $g_i^{\text{PC}} \cdot g_j < 0$ **then** ▷ Indicating conflict
        Project g_i^{PC} onto the normal plane of g_j:
        $g_i^{\text{PC}} = g_i^{\text{PC}} - \frac{g_i^{\text{PC}} \cdot g_j}{\|g_j\|^2} g_j$
    end if
end for
Sum up the projected gradients to get the update $\Delta\theta = \sum_i g_i^{\text{PC}}$

4.5 Including a Separator

To enhance the model's ability to handle multiple tasks, we include a special separator token that helps distinguish between different types of input sequences.

- Separator Token: A unique token is inserted between different segments of the input to clearly demarcate boundaries, helping the model to better understand and process complex inputs.

5 Experiments

5.1 Data

Our experiments utilized three primary datasets:

- **Stanford Sentiment Treebank (SST) Dataset**: This dataset includes 11,855 single sentences extracted from movie reviews and labeled as negative, somewhat negative, neutral, somewhat positive, or positive, allowing for 5 distinct labels from 0-4.
- **Quora Question Pairs Dataset**: This dataset features 404,301 question pairs with binary labels ("Yes" or "No") to denote the status of questions (if a pair of sentences are paraphrases).
- **SemEval Semantic Textual Similarity (STS) Benchmark Dataset**: This dataset contains 8,628 sentence pairs rated for similarity. A 0 indicates that a pair is not at all related, whereas a 5 indicates equivalent meaning.

5.2 Evaluation Method

To evaluate the performance of our models, we used the following metrics:

- **Sentiment Classification (SST)**: Accuracy, based on the proportion of correctly predicted labels.
- **Paraphrase Detection (Quora)**: Accuracy, based on the number of correctly identified paraphrase labels.
- **Semantic Textual Similarity (STS)**: Pearson correlation coefficient between predicted similarity scores and actual labels, reflecting alignment with human judgment.

5.3 Experimental Details

We explored various fine-tuning techniques, evaluating each model's performance across the SST, Quora, and STS datasets. The key experimental setups included:

Baseline Setup For the baseline minBERT implementation:

- **Optimizer and Learning Rate**: We used the ADAM optimizer. When BERT parameters were frozen, the learning rate was 1×10^{-3}. When updating BERT parameters, the learning rate was 1×10^{-5}.

- **Batch Size and Epochs**: Batch size was set to 8 with a 0.3 dropout rate, over 10 epochs.

- **Pre-Training and Fine-Tuning**: We evaluated the development set accuracies obtained during pre-training to determine the effectiveness of each approach.

Extended Setup For the extensions involving PCGrad and schedule optimization:

- **PCGrad Implementation**: PCGrad was used to address gradient conflicts in multi-task learning, adjusting gradients to improve training efficiency and task performance.

- **Schedule Optimization**: We employed round-robin scheduling to balance training across tasks and dynamic batch allocation to ensure challenging tasks received more attention.

Advanced Techniques For integrating cosine embedding loss and Siamese/triplet network structures:

- **Cosine Embedding Loss**: Integrated to optimize training for semantic similarity tasks, capturing nuanced relationships.

- **Siamese/Triplet Networks**: Used to generate effective embeddings, improving feature extraction and representation.

6 Results

Table 1: Training and Dev Accuracies Across Epochs (Last-Linear-Layer)

Epoch	Train Acc SST	Dev Acc SST	Train Acc STS	Dev Acc STS	Train Acc Quora	Dev Acc Quora	Model Acc
1	0.4188	0.3924	0.3819	0.3500	0.6912	0.6932	0.5869
2	0.4273	0.4005	0.4445	0.4308	0.7018	0.7025	0.6061
3	0.3895	0.3787	0.4491	0.4385	0.7249	0.7242	0.6074
4	0.4407	0.3951	0.4625	0.4401	0.7302	0.7281	0.6144
5	0.4199	0.3987	0.4739	0.4491	0.7230	0.7228	0.6154
6	0.4608	0.4078	0.4900	0.4721	0.7294	0.7282	0.6240
7	0.4635	**0.4187**	0.4779	0.4585	0.7336	0.7324	0.6268
8	0.4357	0.3996	0.4957	0.4838	**0.7463**	**0.7444**	0.6286
9	**0.4696**	0.4133	0.4925	0.4826	0.7424	0.7386	0.6310
10	0.4485	0.4133	**0.5045**	**0.4992**	0.7461	0.7436	**0.6355**

Table 2: Training and Dev Accuracies Across Epochs (Full-Model)

Epoch	Train Acc SST	Dev Acc SST	Train Acc STS	Dev Acc STS	Train Acc Quora	Dev Acc Quora	Model Acc
1	0.4775	0.4214	0.8039	0.7568	0.8371	0.8158	0.7052
2	0.5877	0.4995	0.9035	0.8267	0.8769	0.8284	0.7471
3	0.6572	0.4832	0.9357	0.8305	0.9169	0.8459	0.7481
4	0.7307	0.5014	0.9554	0.8363	0.9405	0.8479	**0.7558**
5	0.7906	**0.4959**	0.9632	0.8323	0.9650	0.8562	0.7561
6	0.8329	0.4823	0.9719	**0.8363**	0.9738	0.8512	0.7505
7	0.8649	0.4868	0.9754	0.8288	0.9840	0.8583	0.7532
8	0.8754	0.4714	0.9790	0.8261	0.9878	0.8550	0.7465
9	0.9181	0.4859	0.9809	0.8287	0.9904	**0.8584**	0.7529
10	**0.9306**	0.4759	**0.9836**	0.8336	**0.9932**	0.8558	0.7495

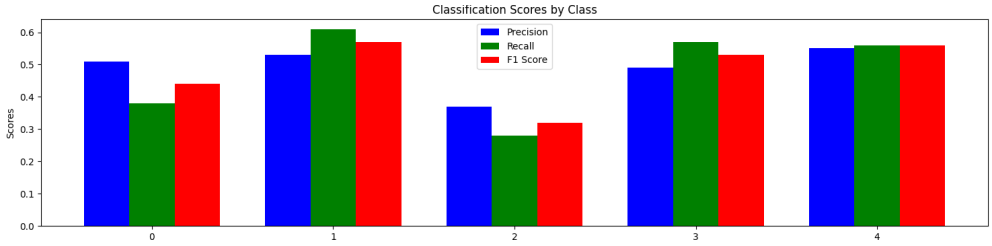

Figure 1: Classification scores by classes in Confusion Matrix.

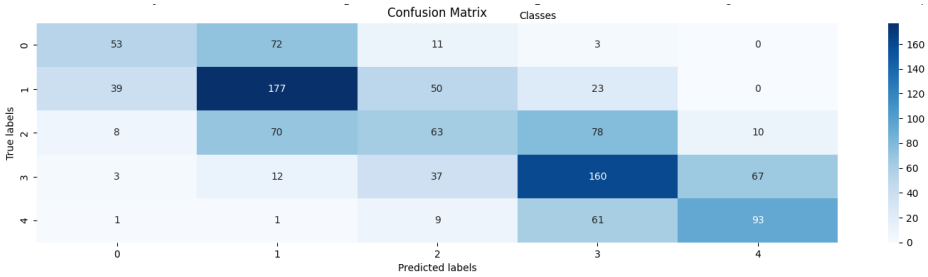

Figure 2: Confusion Matrix showing the model's performance across different classes.

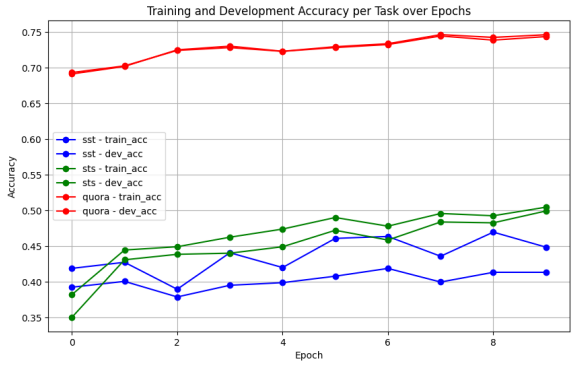

Figure 3: Training accuracies over epochs for different tasks during the pre-train phase.

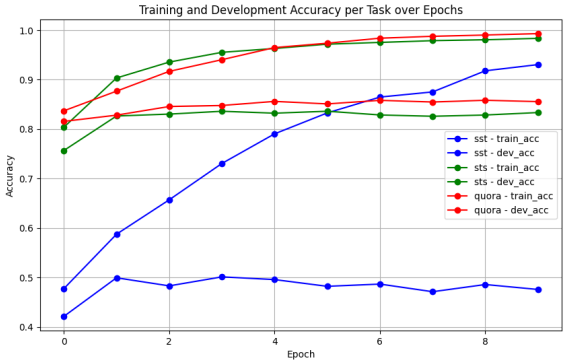

Figure 4: Training accuracies over epochs for different tasks during the fine-tune phase.

7 Analysis

Significant trends can be observed across training and development data collected over the last linear layer (pre-training) and the full model (fine-tuning) that prove to be invaluable to contextualizing the model's learning dynamics. Task performance for STS, SST, and Quora over 10 epochs are clearly illustrated by pre-train accuracies. For sentiment analysis (SST), a fluctuating pattern can be observed that indicates possible instability in the model's learning process. Although it attempts to learn the task, it seems to struggle to maintain consistent performance as underscored by the initial increase preceding the aforementioned fluctuations. The relative stability of development accuracy (with slight increases), on the other hand, shows consistent, although limited, generalization capability. Our semantic textual similarity task's (STS) improvement in development accuracy at a slower rate when compared to training accuracy suggests that even though the model is benefitting from pre-training, overfitting can be an issue as the gap between the two sets of data widen over time. Paraphrasing detection (Quora), however, exhibits a strong upward trend that is visible across both development and training, which indicates effective learning and generalization, underscored by the observation that development nears 0.75 by the final epoch. Fine-tuning demonstrates significant observable improvement across each task, with each process seeming to significantly improve the model's ability to capture task nuance in sentiment analysis (SST), for example. A sharp increase is seen during training, reaching around 0.8 by the final epoch. Not only that, but semantic textual similarity (STS) also shows significant accuracy improvements, reaching nearly 1.0. Even more promising, development accuracy for STS increases substantially as well, highlighting effective generalization. For our Quora task, robust learning and effective generalization are observed through similar trends as well. The experiments' resulting confusion matrix, however, reveals possible issues. Class 1 demonstrated the maximum accuracy observed, with the majority of true labels correctly predicted (177 out of 289). However, classes 0 and 2 saw significant misclassifications where many instances appeared to be confused with class 1, indicating that the model struggles to distinguish features unique to these classes. Class 4 exhibits the minimum number of classifications, which suggests the presence of distinct features that the model can identify quite easily. Overall, model accuracy is 0.50, with a macro-average F1 score of 0.48. This reflects moderate performance and room for improvement, especially in reducing misclassification and enhancing feature discrimination across the various classes.

7.1 PCGrad

Results show trends of overfitting, especially when it comes to the sentiment analysis task (SST) during the fine-tuning phase. Gaps seen between training and development data accuracies spotlight the model's ability to learn training data well, but also the trend that it seems to struggle greatly to generalize unseen data. A major contribution to overfitting appears to be the use of PCGrad with limited data, which leads to observed degradation of performance on the development set. Following further analysis, we recommend training for fewer epochs as an effective solution to overfitting, with an optimal number of 4 epochs providing sufficient learning while also maximizing pattern recognition. Given that PCGrad combined with limited data contributes significantly to overfitting, the use of PCGrad must also be re-evaluated in order to adjust gradient surgery parameters or combine PCGrad with additional regularization.

7.2 Schedule Optimization

Round-robin scheduling's ability to balance training across tasks resulted in a significant positive impact on model performance with steady improvements visible in training and development accuracies across SST, STS, and Quora. Paraphrase detection (Quora) especially showed robust learning and effective generalization, with development accuracy consistently above 0.9 in the fine-tuning phase, clearly showing hte impact of dynamic batch allocation and balanced task scheduling in allocation resources and attention, as well enhancing generalization. The dynamic nature of this scheduling also ensured that the more challenging tasks received focus accordingly and improved performance.

7.3 Cosine Embedding Loss

The integration of Cosine Embedding Loss proved crucial for improvin the performance of the semantic textual similarity task (STS) as both training and development accuracies approached 1.0 in

fine-tuning. This shows that cosine embedding loss effectively captured semantic relationships and improved generalization. On top of that, the high accuracy indicates that it is incredibly effective for tasks that require a more nuanced understanding of semantic similarities. Its ability to dynamically adjust margins contributes greatly to the model's performance when it comes to differentiating between similar and dissimilar pairs..

7.4 Siamese and Triplet Network Structures

The goal behind the incorporation of Siamese and triplent network structures was to generate embeddings that reduce the time behind similarity calculations in semantic tasks, and was inspired by Sentence-BERT. As a result, significant improvements were observed to tasks like paraphrase detection (Quora), which is highlighted by the strong upward trends as the final epoch was approached. It generated more effective embeddings that lead to improved accuracy and provided structural enhancement that directly enabled the model to more effectively understand and process semantic input.

8 Conclusion

Ultimately, our results demonstrate that careful application and adjustment of PCGrad, Cosine Embedding Loss, Siamese and triplet network structures, and round-robin scheduling leads to substantial improvements in model performance and generalization. Trends of over-fitting, however, were observed with PCGrad application when combined with limited data. Primarly limitations include include computation resources, funding, as well as memory and RAM availability. For future work, we aim to employ Principal Component Analysis (PCA) for dimensionality reduction to streamline computational efficiency and enhance the semantic precision of our embeddings.

9 Ethics Statement

What are the ethical challenges and possible societal risks of your project, and what are mitigation strategies?

Enhancing miniBERT's capabilities by adding additional neural network layers introduces significant ethical challenges and societal risk. We will explore these risks primarily through the lenses of two guiding principles: transparency and accountability. As the complexity of the system increases, the transparency of the model is reduced, making it harder to interpret and understand its decision making. This lack of transparency can lead to accountability issues in critical applications. In addition, the more complex model might inadvertently learn and perpetuate biases present in training data that lead to discriminatory outcomes in areas such as hiring, lending, and law enforcement. Not only that, but the increased computation requirements needed to train deeper networks raise concerns regarding environmental sustainability due to the large energy consumption. In order to mitigate these risks, we recommend a few strategies. Thorough audits must be conducted of the training data and model outputs in order to identify and address biases early while engaging with a diverse group of stakeholders. This includes ethicists and affected communities to ensure the input of various perspectives. In addition, ethical practices must be promoted within development team culture on top of optimizing resource use.

Our optimization extensions also add another more layers of considerations. For example, the scheduling extension uses dynamic batch allocation and task scheduling, which in turn raise ethical questions around allocation efficiency and performance. Even though the round-robin approach attempts to balance cross-task training, the risk that computation resources may not be optimally utilized still exists, which leads to inefficiencies and increased costs. Dynamic batch allocation could also cause performance inconsistencies that affect the overall model. On top of that, as the number of tasks increases, scalability and maintenance issues also increase and complicate the scheduling process. To mitigate this, optimizing resource allocation must be prioritized and consistent performance ensured across tasks. This can be more specifically done through the regular monitoring of resource usage and task performance, engaging with stakeholders, and fostering a culture of sustainability. Separately as well, cosine embedding loss extension raises ethical concerns related to similarity measures and marginalization of less common data points, which must also be mitigated. By only relying on cosine similarity, important contextual differences might also be overlooked leading

to harmful outcomes in applications like recommender systems or social media. This approach could marginalize outliers and affect the model's inclusivity and diversity. Ensuring that similarity measures are appropriate for the application context and incorporate diverse perspectives is a great first step. By thoroughly validating these similarity measures and embedding space, it becomes possible to capture relevant distinctions without marginalizing outliers. On top of that, it is crucial to engage domain experts and affected users as their feedback is incredibly valuable regarding the effectiveness and fairness of the embedding space and combine that with regular updates and adjustments to the loss function. In terms of the gradient surgery (PCGrad Rule) extension, some of the most important concerns are overfitting and underfitting. The accompanying complexity also has the potential to reduce interpretability and transparency of the training process. These risks can be mitigated through continuous monitoring of the impact on model performance and validating the approach by speaking with end users. Accountability and transparency can be enhaced even more through robust validation and testing procedures that combine clear documentation and visualization of the gradient surgery process. Lastly, our inclusion of a separator to handle multiple tasks poses data integrity concerns. The use of separators must be carefully managed to avoid corrupting the input data or altering its intended meaning. One way this can be mitigated is through the auditing of input data to validate data integrity.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[2] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

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

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

[5] Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, 2020.

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

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[8] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[9] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc., 2020.