minBERT and Multi-Task Learning for Downstream Tasks

Stanford CS224N Default Project
Mentor: Candice Penelton. No external collaborators, not sharing project.

Michael Zhu
Department of Computer Science
Stanford University
mszhu@stanford.edu

Jonathan Coronado
Department of Computer Science
Stanford University
jcoronad@stanford.edu

Abstract

The Bidirectional Encoder Representations from Transformers model (BERT) proposed by Devlin et al [1] significantly improved fine-tuning based, pre-trained approaches to downstream NLP tasks. In this project, we implement key features of BERT and go further to implement extensions to pretrained BERT to further improve BERT's performance on multiple downstream tasks. Implemented extensions augment BERT's finetuning mechanisms to train the model for three tasks – sentiment classification, semantic textual similarity, and paraphrase detection – by altering how and when the model is trained for certain tasks and or by using a different objective function. Further experimentation was also conducted with the model's hyperparameters. We find that a combination of Gradient Surgery, as described by [2] and Pearson Correlation Loss resulted in the best performance both task-wise and on average with a improvement in average accuracy by 0.24 over all three tasks. However, other permutations of extensions were also shown to on average improve upon base BERT's performance with an average accuracy of 0.453 versus base BERT's 0.299. It was also found that the model generally performs better with a batch size of 8 and learning rate of $1e{-}5$. Our findings suggest that Gradient Surgery is a viable method for multitask finetuning and further benefits from the use of Pearson Correlation Loss as its objective function, though other permutations are also potentially promising.

1 Introduction

The Bidirectional Encoder Representations from Transformers (BERT) model was originally proposed by Devlin et al in 2018 and was the first pre-trained language model to generate deep, bidirectional representations with the novel Masked Language Model (MLM) and Next Structure Prediction (NSP) pre-training tasks [1]. BERT outperformed all existing models on eleven natural language processing tasks, including GLUE and SQUAD, and was the first fine-tuning representation model to achieve state-of-the-art performance on both sentence-level and token-level tasks. BERT was transformative to the field of natural language processing, illustrating the power of pre-trained, transformer-based models with simpler, shared architectures across tasks.

While BERT was originally pre-trained with unsupervised learning and then fine-tuned on individual tasks, the data and computing requirements of individually-trained methods make it difficult to learn multiple tasks. An intuitive approach would be to train a model on all tasks simultaneously, which could potentially generate a shared architecture across tasks that achieves greater efficiency and performance. However, learning multiple tasks at once is a difficult optimization problem, and multi-learning sometimes results in worse overall performance and data efficiency compared to learning tasks individually [3]. Consequently, there has been much research into solving the multi-task learning problem. For instance, Cooper and Murray propose a novel method of multi-task learning with BERT, sharing information between tasks and reducing the number of parameters required [4].

They use novel projected attention layers to match performance of individually fine-tuned models on the GLUE benchmarks with 7 times fewer parameters. There has been much literature focused on finding architectural solutions to the multi-task learning problem, including Liu et al [5], who propose an attention-based multi-task learning architecture, and Vandenhende and Georgoulis [6] who propose an approach to automatically construct branched multi-task networks. Others have tried to generate algorithmic solutions to the multi-task learning problem. For instance, Yu et al proposed a technique called Gradient Surgery that prevents conflicting gradients in training, showing that such a method leads to improved multi-learning [2].

The starting point for this project is the base minBERT model provided by CS 224N, which is a smaller version of the BERT model. We implemented several key features of the minBERT model, as well as the step() function of the Adam optimizer. Our project's goal is to fine-tune and extend the minBERT model to successfully perform well on multiple sentence level tasks simultaneously - sentiment classification, paraphrase identification, and semantic textual similarity evaluation.

The base minBERT is only fine-tuned on the sentiment classification task and then applied to all three tasks, which clearly is insufficient. To remedy this issue and improve BERT's performance on all three three downstream tasks, we implement various extensions to base minBERT. In particular, we look at "Round Robin" multi-task fine-tuning, multi-task fine-tuning with Gradient Surgery, Cosine Similarity Learning, and Pearson Correlation Loss. We experiment with and evaluate various combinations of these extensions, and we show that each of these extensions improves performance from base minBERT. In particular, Gradient Surgery and Pearson Correlation Loss significantly improve the model performance.

2 Related Work

2.1 BERT and its iterations

Since BERT was originally proposed in 2018, many iterations have followed. Yang et al proposed XLNet, a large bidirectional transformer model, that improved upon BERT by introducing a permutation language model, where all tokens are predicted in a random order rather than a sequential order [7]. XLNet performed better than BERT on 20 different language tasks. Around a similar time as XLNet, Liu et al. proposed RoBERTa, a robustly optimized BERT approach that re-trained BERT with 1000% more data [8]. RoBERTa also removed the Next Sentence Prediction task from its pre-training and used dynamic masking during training. RoBERTa outperformed both BERT and XLNet on the GLUE tasks. Others have also sought not to create better performing versions of BERT, but faster versions. Sanh et al proposed DistilBERT, a smaller version of BERT with half the parameters but that retains 95% performance, which is useful for faster inference speed [9]. There also exist Lan et al's ALBERT model, which is a lighter version of BERT [10], and Clark et al.'s ELECTRA model, which replaces the MLM task with a Replaced Token Detection task and also removes the NSP task [11]. This summary of BERT improvements is non-exhaustive, and the field of pre-trained language models constantly continues to iterate and find novel improvements.

2.2 Applications of BERT to Sentiment Classification, Paraphrase Detection, and Semantic Textual Similarity Classification

Previous literature has also sought to apply BERT to the downstream tasks of Sentiment Classification, Paraphrase Detection, and Semantic Textual Similarity Classification. Xu et al. extended BERT by training it on a novel dataset for aspect-based sentiment analysis as well as adding an extra task-based layer [12]. They were able to achieve a max accuracy of 84.26% with their extended model. Ko and Choi introduced a novel fine-tuning approach for BERT to learn representations for sentence-level paraphrase identification [13]. They first fine-tune BERT on a variety of NLP tasks, including GLUE tasks and question answering tasks, and then fine-tune BERT on the target paraphrase identification task. Their improved model achieved an accuracy of 89.2%. Finally, Yang et al. fine-tuned BERT on a massive corpus of clinical texts, achieving a max Pearson correlation of 0.8615 [14]. In general, Mutinda et al find that state-of-the-art systems achieve Pearson correlations of over 0.80 on STS tasks [15].

3 Approach

3.1 minBERT

Our baseline model is the provided minBERT model, which is a minimalist implementation of the BERT model from Devlin et al [1]. We also use the Adam optimizer with Decoupled Weight Decay Regularization as our optimizer. This model and the Adam optimizer are described in the Default Project Handout. We implemented specific key features of the minBERT model and the step() function of the Adam optimizer, all of which are also described in the Default Project Handout.

The base multitask classifier of the minBERT model is only fine-tuned on the provided SST train dataset. There are three heads, one for each task (Sentiment Classification, Paraphrase Detection, STS), attached on top of these minBERT embeddings. For the SST head, the input ID and attention mask are passed through the pre-trained BERT model. The pooler output BERT is then passed through a ReLU activation layer, then a dropout layer with a dropout probability of 0.3, and then a linear layer returning a final logit. The paraphrase detection and STS heads follow a similar structure, except the BERT pooler outputs from the separate input IDs and attention masks are concatenated before being passed through the rest of the layers. The heads are illustrated below.

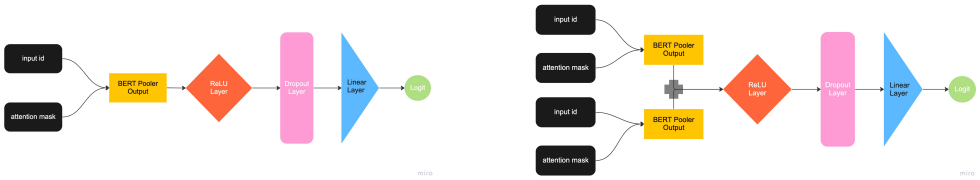

Figure 1: Sentiment Classification Head (Left) and STS and Paraphrase Detection Heads (Right)

3.2 minBERT Extensions

We extend minBERT by augmenting the multitask classifier to perform multi-task fine-tuning on all three tasks' training datasets. We initially implemented a simple Round-Robin training approach, where we sequentially fine-tuned the BERT parameters on all three training datasets. This Round-Robin structure is illustrated in the figure below.

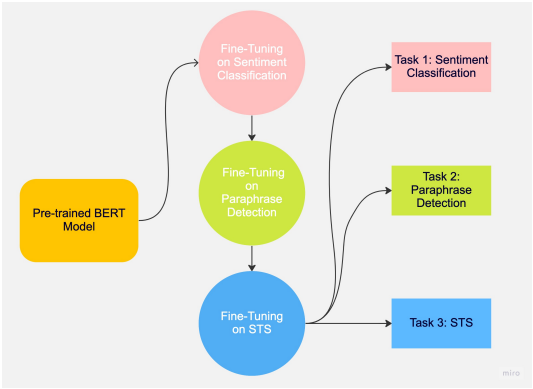

Figure 2: Round-Robin Multi-Task Fine-tuning

To further augment the multi-task fine-tuning, we also implement gradient surgery, as described in [2]. Gradient Surgery projects the gradient of the i-th task $\mathbf{g}_i$ onto the normal plane of another conflicting task's gradient $\mathbf{g}_j$, performing the following update.

$$\mathbf{g}_i = \mathbf{g}_i - \frac{\mathbf{g}_i \cdot \mathbf{g}_j}{||\mathbf{g}_j||^2} \cdot \mathbf{g}_j \tag{1}$$

The algorithm for gradient surgery is described below by Yu et al [2].

Require: Model parameters θ, task minibatch $\mathcal{B} = \{\mathcal{T}_k\}$

$\mathbf{g}_k \leftarrow \nabla_\theta \mathcal{L}_k(\theta) \ \ \forall k$

$\mathbf{g}_k^{\text{PC}} \leftarrow \mathbf{g}_k \ \ \forall k$

for $\mathcal{T}_i \in \mathcal{B}$ **do**

  for $\mathcal{T}_j \overset{\text{uniformly}}{\sim} \mathcal{B} \setminus \mathcal{T}_i$ in random order **do**

    if $\mathbf{g}_i^{\text{PC}} \cdot \mathbf{g}_j < 0$ **then**

      // Subtract the projection of $\mathbf{g}_i^{\text{PC}}$ *onto* $\mathbf{g}_j$

      Set $\mathbf{g}_i^{\text{PC}} = \mathbf{g}_i^{\text{PC}} - \frac{\mathbf{g}_i^{\text{PC}} \cdot \mathbf{g}_j}{\|\mathbf{g}_j\|^2} \mathbf{g}_j$

return update $\Delta\theta = \mathbf{g}^{\text{PC}} = \sum_i \mathbf{g}_i^{\text{PC}}$

Figure 3: PCGrad Update Rule

We use Tseng's Pytorch implementation of PCGrad to wrap our Adam optimizer and simultaneously train on all three tasks' training datasets [16].

We also implement Cosine-Similarity fine-tuning, specifically for semantic textual similarity evaluation. In this setup, cosine-similarity between two embeddings x_1 and x_2 is calculated as follows, where $\epsilon = 1\text{e}{-8}$ is used to avoid division by zero.

$$similarity = \frac{x_1 \cdot x_2}{max(||x_1||_2||x_2||_2, \epsilon)} \tag{2}$$

We also experimented with using Pearson correlation loss for fine-tuning on the STS task, given that Pearson correlation is the measurement generally used for STS evaluation and could potentially be a better measurement of "loss" than basic cross entropy loss. The equation for our Pearson correlation loss function is below.

$$loss = 1 - r = 1 - \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 (y_i - \bar{y})^2}} \tag{3}$$

Finally, we perform some hyper-parameter optimization. We perform experiments with varying values for batch size and learning rate.

4 Experiments

4.1 Data

We are using the provided datasets, which are described in the Default Project Handout. For sentiment analysis, we will be using two datasets: the Stanford Sentiment Treebank (SST) dataset and the CFIMDB dataset. The dataset contains a total of 215,154 unique phrases from 11,855 single sentences from movie reviews, and each phrase is labeled negative, somewhat negative, neutral, somewhat positive, or positive. The CFIMDB dataset consists of 2434 movie reviews (many of which are longer than just one sentence), each of which has a binary label of negative or positive. For paraphrase detection, we will be using the Quora Dataset, which contains 400,000 question pairs with labels indicating whether the pairs are paraphrases of each other. For semantic textual similarity, we will be using the SemEval STS Benchmark Dataset, which consists of 8628 sentence pairs of varying similarity on a scale from 0 (unrelated) to 5 (equivalent meaning). We are provided with training, development, and testing splits, so we will be using those provided splits.

4.2 Evaluation method

We will use accuracy (percentage correct) as our evaluation metric for performance on the sentiment classification and paraphrase detection tasks. For the semantic textual similarity evaluation task, we will be using the Pearson correlation of the true similarity values against the predicted similarity values across the test dataset.

4.3 Experimental details

We ran our experiments with various combinations of the features described in Section 3.2. For our base experiments, we used a learning rate of $1e{-}5$, 10 training epochs, a batch size of 8, and a dropout probability of 0.3 for dropout layers. We also kept BERT's default hidden size of 768. The objective function, unless stated otherwise, is cross entropy loss (or binary cross entropy loss for binary labels).

4.4 Results

Table 1: Summary of models' performance on dev examples

Experiment	Sentiment Accuracy	Paraphrase Detection Accuracy	STS Pearson Correlation	Average Score
minBERT	0.514	0.435	-0.053	0.299
minBERT + Round Robin	0.51	0.474	-0.013	0.324
minBERT + Round Robin + Cosine Similarity	0.49	0.521	0.247	0.419
minBERT + Gradient Surgery	0.458	0.625	0.038	0.378
minBERT + Gradient Surgery + Cosine Similarity	0.517	0.717	0.325	0.519
minBERT + Gradient Surgery + Cosine Similarity (no ReLU layers)	0.522	0.71	0.223	0.485
minBERT + Gradient Surgery + Pearson Correlation Loss + Cosine Similarity	0.506	0.73	0.29	0.509
minBERT + Gradient Surgery + Pearson Correlation Loss	**0.525**	**0.734**	**0.357**	**0.539**

We were able to substantially improve upon the baseline minBERT model. Particularly, we saw the most improvement within the paraphrase detection and STS evaluation tasks, with increases of around 30% and .41, respectively.

Unsurprisingly, both Round Robin and Gradient Surgery multi-task fine-tuning led to better performances on the paraphrase evaluation task and STS evaluation task, with Gradient Surgery having a greater improvement than Round Robin.

Furthermore, both Cosine Similarity fine-tuning and Pearson Correlation loss led to improvements within the STS evaluation task. We were not surprised that Pearson Correlation loss had greater improvements than Cosine Similarity fine-tuning, given that Pearson Correlation was the evaluation metric for final testing and thus would be a better measurement of similarity than cosine-similarity. However, we were surprised that these extensions, which were only implemented for STS, improved our model's performance on the paraphrase detection task as well. Similarly, we were also surprised that the combination of both Cosine Similarity and Pearson Correlation loss actually hurt performance as opposed to Pearson Correlation loss alone.

While we did not expect significant improvements in sentiment classification given we did not implement any sentiment-specific extensions, we thought that fine-tuning on the other tasks could improve performance on sentiment as well, so we were surprised that our model's performance on sentiment classification did not change much.

Finally, we also experimented with the multi-task heads' architecture, removing the ReLU activation function, which was originally added to prevent vanishing gradients, to see if performance changed at all. We saw that the ReLU activation function was indeed useful, and performance was hurt when it was removed.

Overall, our final model saw substantially better performance than the base minBERT model on the dev examples, with an overall score improvement of 0.24.

We also performed hyperparameter optimization on the final model (minBERT + Gradient Surgery + Pearson Correlation Loss), varying batch sizes and learning rates. The results are shown below.

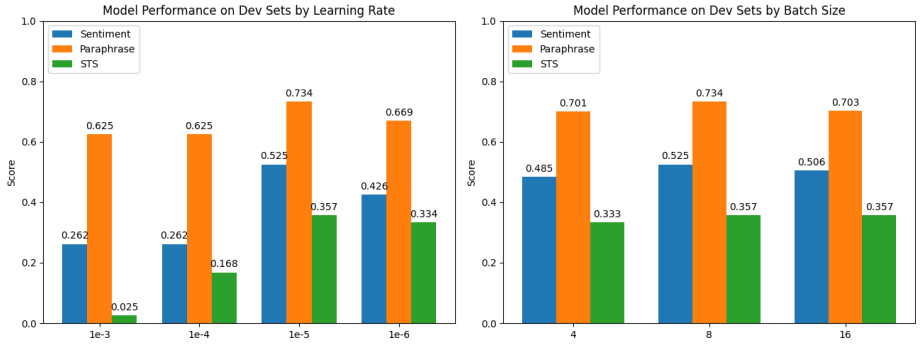

Figure 3: Model Performance by Learning Rate (left) and Model Performance by Batch Size (right)

Surprisingly, the default settings of a learning rate of $1e{-}5$ and a batch size of 8 lead to the best overall results.

The larger learning rates $1e{-}3$ and $1e{-}4$ often led to exploding gradients for the STS loss function during training, which worsened performance. The smaller learning rate of $1e{-}6$ potentially led to more overfitting on the train sets, which worsened performance on the dev sets.

It has been observed that larger batch sizes lead to significant degradation in model quality [17], so it is surprising that a batch size of 8 performs better than a batch size of 4. The models perform fairly similar, so the difference could be simply from randomness. It is also possible that the batch size has interactions with the learning rate that we did not explore here. Particularly, Kandel and Castelli showed that when learning rates are high, the large batch size performs better than with small learning rates [18]. Overall, further testing and evaluation is required to discover the optimal batch size, but we proceed with the batch size of 8.

Our final model achieved the results below on the test leaderboard.

Table 2: Final model's performance on test leaderboard

Experiment	SST Accuracy	Paraphrase Detection Accuracy	STS Pearson Correlation	Overall Score
minBERT + Gradient Surgery + Pearson Correlation Loss (Batch Size = 8 and Learning Rate = 1e-5)	**0.520**	**0.734**	**0.357**	**0.537**

5 Analysis

5.1 Gradient Analysis

From the previous section, we see that Gradient Surgery multi-task fine-tuning leads to greater improvements than Round Robin multi-task fine-tuning. This is unsurprising, given that Round Robin multi-task fine-tuning is likely to run into conflicting gradient updates between tasks, which undermines the overall effectiveness of multi-task fine-tuning. Similarly, upon manual examination of the gradients resulting from Gradient Surgery, we saw that the STS loss gradients often became exploding gradients, which prevented our model from generating effective parameters. Gradient Surgery remedies these gradient issues, leading to greater success across tasks.

5.2 Model Failures

Next, we analyze when our model fails, segmented by task.

1. Sentiment Classification

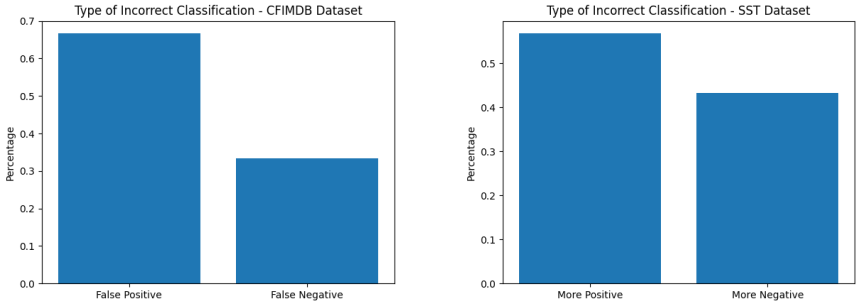

Figure 4: Model Failure on Sentiment Classification by Failure Type

For sentiment classification, when our model made an incorrect prediction, it tended to incorrectly classify a sentence as positive. An example of when our model made such a false positive prediction is below.

This riveting World War II moral suspense story deals with the shadow side of American culture: racial prejudice in its ugly and diverse forms .

This example has a nuanced sentiment, recognizing positive elements of a story but overall expressing negative sentiment about American culture. However, the model fails to pick up these nuances, and the model's positive prediction seems reasonable for the first half of the quotation. Consequently, we suspect that the model is too sensitive to positive words, and to improve the model, we could fine-tune the model more on negative words.

2. Paraphrase Detection

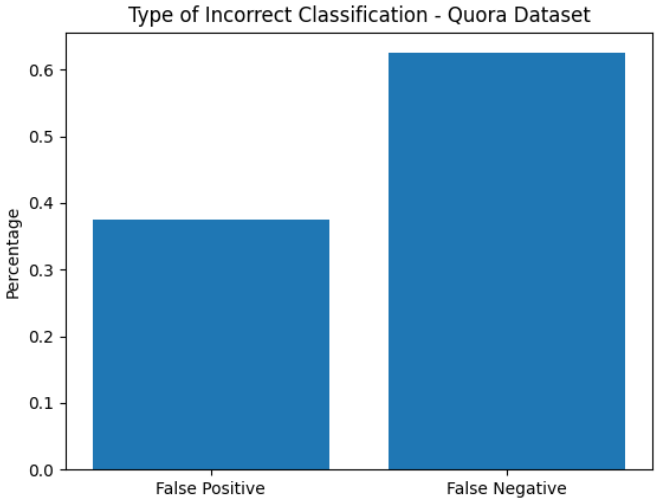

Figure 5: Model Failure on Paraphrase Detection by Failure Type

We see that our model is more likely to false designate pairs of paraphrases as not paraphrases. Two examples of this false negative behavior are shown below.

Sentence 1: Why are Facebook, Google, and others not allowed in China? **Sentence 2:** Why are Google, Facebook, YouTube and other social networking sites banned in China?

Sentence 1: What makes one angry? **Sentence 2:** What is the one thing that makes you most angry?

In both pairs of sentences, there are specific modifications that could confuse our model. In the first pair, the addition of Youtube and "social networking sites" could lead to different interpretations by our model, and in the second pair, "the one thing" could cause our

model to interpret the second question differently. Overall, the threshold for our model making a positive determination seems too high, and we could look into ways for weighting similarities more.

3. Semantic Textual Similarity

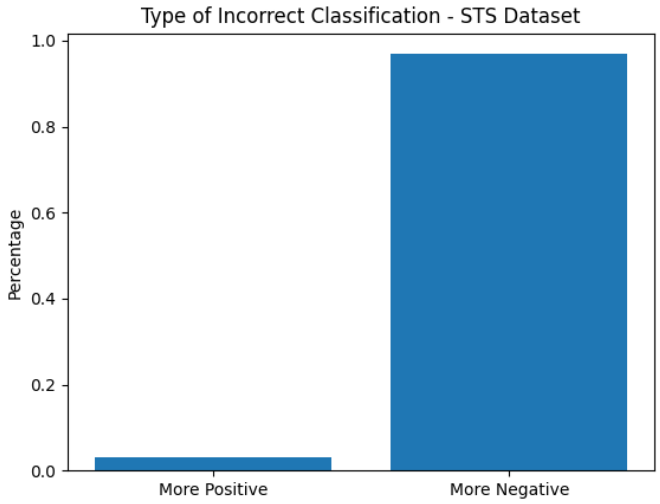

Figure 4: Model Failure on Semantic Textual Similarity by Failure Type

Our model seems to always designate sentences' similarity as more negative than they actually area. Similar to paraphrase detection, the threshold for our model making a positive determination seems too high, and we could weight similarities more.

6 Conclusion

In this project, we implement and explore several optimizations to BERT for downstream multitask performance. The optimizations explored were Gradient Surgery for multitask finetuning as described by [2] and implemented by [16], "Round Robin" multitask finetuning, Cosine-Similarity finetuning, and Pearson Correlation loss. We also experimented with altering the model's architecture and hyperparameters, specifically batch size learning rate, and the inclusion or exclusion of ReLU layers.

All implemented extensions were shown to improve the model's performance, though the combination of Gradient Surgery and Pearson Correlation Loss was shown to most significantly improve performance with accuracies of 0.525 on sentiment classification, 0.734 on paraphrase detection, and 0.357 on semantic textual similarity versus the base model's 0.514, 0.435, and -0.053 respectively, all when evaluated on the dev set. Similar results were achieved on the test set with marks of 0.520 on sentiment classification, 0.734 on paraphrase detection, and 0.357 on Semantic Textual Similarity. When experimenting with the model's architecture, we found that as expected, the model performed worse without the ReLU activation layers. When adjusting the hyperparameters, we found that a learning rate of $1e{-}5$ and a batch size of 8 allowed for optimal performance. This result was somewhat surprising as previous research has shown that smaller batch sizes could perform better than larger sizes [17].

Despite these successes, there are some limitations of our model which should be noted. Notably, out of all tasks, our model made the least improvement in sentiment classification, with an accuracy increase of only 0.011 in comparison to marks of 0.299 and 0.410 for paraphrase detection and STS correlation respectively. Additionally, our model only focuses on three downstream tasks and it is unclear how it would perform with additional tasks. Next steps to further improve the model could include training for additional tasks as well as further finetuning to improve Sentiment Classification performance.

References

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

[2] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

[3] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015.

[4] Asa Cooper Stickland and Iain Murray. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. *arXiv e-prints*, pages arXiv–1902, 2019.

[5] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019.

[6] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. *arXiv preprint arXiv:1904.02920*, 2019.

[7] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[9] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[10] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[11] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[12] Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*, 2019.

[13] Bowon Ko and Ho-Jin Choi. Paraphrase bidirectional transformer with multi-task learning. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 217–220. IEEE, 2020.

[14] Xi Yang, Xing He, Hansi Zhang, Yinghan Ma, Jiang Bian, Yonghui Wu, et al. Measurement of semantic textual similarity in clinical texts: comparison of transformer-based models. *JMIR medical informatics*, 8(11):e19735, 2020.

[15] Faith Wavinya Mutinda, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. Semantic textual similarity in japanese clinical domain texts using bert. *Methods of Information in Medicine*, 60(S 01):e56–e64, 2021.

[16] Wei-Cheng Tseng. Weichengtseng/pytorch-pcgrad, 2020.

[17] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[18] Ibrahem Kandel and Mauro Castelli. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT express*, 6(4):312–315, 2020.