Multi-BERT: A Multi-Task BERT Approach with the Variation of Projected Attention Layer

Stanford CS224N Default Project

Haijing ZHANG
Department of Electrical Engineering
Stanford University
haijing@stanford.edu

Abstract

This project delves into sophisticated techniques for fine-tuning the BERT model, aiming to boost its effectiveness across three specific sentence-level tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. The core objective is to refine BERT's embeddings for efficient multi-task learning, ensuring that the performance of individual tasks remains unimpacted. We experimented with a variety of pooling strategies, loss-balancing approaches, and parameter adaptation methods to discover an optimal mix that suits the learning goals of each task. Notably, we introduced a hybrid multi-task training curriculum that outperformed both traditional sequential training and naive multi-task training approaches. Our exploration of four unique parameter adaptation strategies identified that a synergy between low-rank and top attention methods yields the best outcomes. The pinnacle of our efforts was achieving an overall accuracy of 0.759 on the test dataset, with task-specific performances of 0.506 on the SST dataset, 0.856 on the Quora dataset, and 0.830 on the STS dataset, indicating a significant advancement in multi-task learning capabilities.

1 Key Information to include

- Custom or Default Project: Default Project
- Mentor: Anirudh Sriram

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, Devlin 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 project 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, and an originally developed multi-task training curriculum, this project 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 0.759 on a composite test dataset. Specifically, we observed scores of 0.506 on the SST dataset (sentiment analysis), 0.856 on the Quora dataset (paraphrase detection), and 0.830 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

Sentence Bert Reimers and Gurevych (2019) introduces Sentence-BERT (SBERT), a modification of the pretrained BERT network using siamese and triplet network structures. This adaptation allows for the efficient computation of semantically meaningful sentence embeddings, significantly reducing the time required for similarity searches from hours to seconds while maintaining high accuracy, and thereby making BERT feasible for semantic similarity search and clustering tasks. In this project, we leverage the SBERT architecture as the task-specific layer for each task.

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 (Devlin 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. Furthermore, Houlsby et al. (2019) propose a variation of PALs, employing slightly different adapter modules that substitute the multi-attention mechanism with an MLP (Multilayer Perceptron). In this project, we have implemented four types of methodologies from these studies and have combined two of them to enhance our baseline method, aiming to effectively optimize multi-task performance.

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.

Several loss-balance techniques have been developed to address this issue. Chen et al. (2018) introduced GradNorm, which normalizes the gradients of each task's loss to prevent any single task from dominating the training process. Uncertainty weighting, developed by Kendall et al. (2018), uses homoscedastic uncertainty to derive a multi-task loss function, allowing optimal balance between regression and classification losses. This project leverages a method called dynamic weight average, introduced by Liu et al. (2019), which dynamically adjusts the weights of different tasks based on their performance or importance during the training process.

In this work, we use these methods as an extension of the baseline method. Besides, we also integrate these methods with other techniques such as a multi-task training curriculum to simultaneously improve the multi-task performance.

4 Approach

Model Architecture The backbone of miniBERT and the sentiment analysis head are identical to those in part one of the default project. Regarding the other two tasks, Reimers and Gurevych (2019) suggested specific architectural frameworks for sentence-pair tasks, designed particularly for classification and regression problems. Influenced by this, I opted to use Figure 1 for the paraphrase detection task, which is a Siamese network structure (Chen and He, 2021) and can be formulated as $W([u, v, |u - v|])$. Given that this is a binary classification problem, I applied a sigmoid function combined with Binary Cross Entropy (BCE) as the loss function. For the sentiment similarity task, I employed Figure 2 for the head architecture, which processes the embeddings of two sentences through two fully connected layers before calculating the cosine similarity. I then applied a ReLU function to convert all negative similarities to zero and multiplied by 5 to scale the output to 0-5. The loss function used for this task is the Mean Square Error (MSE).

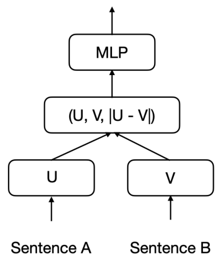

Figure 1: Head of paraphrase detection task.

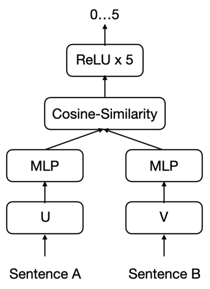

Figure 2: Head of sentiment similarity task.

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.

Parameters Adaption Methods
The concept behind parameter adaptation methods involves freezing the majority of the backbone model, i.e., BERT's parameters, during the fine-tuning process and only tuning task-specific parameters. In addition to the task-specific heads mentioned earlier, I have also implemented four methods based on projected attention, developed by Stickland and Murray (2019) and Houlsby et al. (2019). These were developed to ensure that each task has its own projected attention modules, facilitating the learning of task-specific attention. This approach maintains most parameters as shared, thereby ensuring the model retains a robust common representation and strong generalization capabilities.

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), is depicted in Figure 3. The formulation is as follows:

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

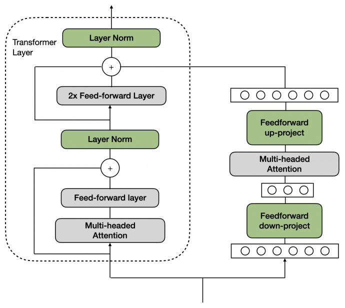

Figure 3: PAL

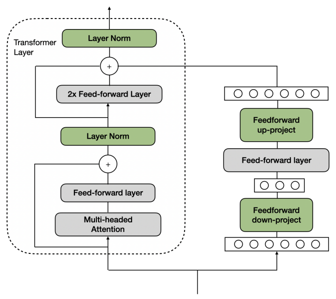

Figure 4: Low Rank

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:

$$TS(h) = V^D\left(MH\left(V^E(h)\right)\right)$$

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, as illustrated in Figure 4. The sole distinction lies in its task-specific layer, TS, which is modified as follows:

$$TS(h) = V^D\left(I\left(V^E(h)\right)\right)$$

Here, the multi-head attention layer is replaced with an identity matrix, transforming this layer into a straightforward low-rank projection.

The third method, known as the Houlsby adapter, is developed by Houlsby et al. (2019) and features a similar task-specific (TS) layer (Figure 5). There are two main differences: firstly, the TS function incorporates a nonlinear layer, specifically a GELU function, between the encoder and the decoder; secondly, it is integrated directly within the transformer layer, rather than being added in parallel. The TS function of the Houlsby adapter is formulated as:

$$TS(h) = V^D\left(GELU\left(V^E(h)\right)\right)$$

The final method involves simply adding a task-specific BERT layer atop the shared BERT model. Its architecture is illustrated in Figure 6.

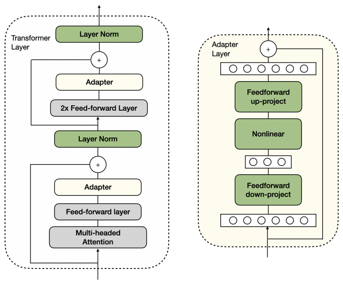

Figure 5: Houlsby Adapter

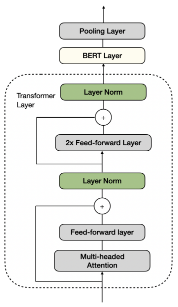

Figure 6: Projected Attention on Top

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. In this study, I compare these four methods along with various combinations of them. The results will be discussed subsequently.

Loss-balance Techniques The loss-balance technique in training BERT is crucial for multitask learning, ensuring that each task's loss is appropriately weighted to prevent dominance by any single

METHOD	PARAMETERS
PAL	$T(2d_m d_s + 12 \times 3d_s^2)$
LOW RANK	$T(12 \times 2d_m d_s)$
HOULSBY ADAPTER	$T(12 \times 4d_m d_s)$
PROJ.ATTEN.ON TOP	$T(12d_m^2)$

Table 1: Total parameters for various parameter adaption methods

task, thereby improving overall model performance and fairness across tasks. So far, I have tried two approaches. The first one is $L = L1^2 + L2^2 + L3^2$ with the simple idea that if any of the loss is too large, then it will use itself to penalize itself. Another approach is the dynamic weight average approach developed by Liu et al. (2019), which can be formulated as:

$$\lambda_k(t) := \frac{K \exp(w_k(t-1)/T)}{\sum_i \exp(w_i(t-1)/T)}, \quad w_k(t-1) = \frac{\mathcal{L}_k(t-1)}{\mathcal{L}_k(t-2)}$$

where $\lambda_k(t)$ is the weight for each task, and $w_k(t-1)$ is the ratio of the loss in the previous round to the round before that, representing the learning rate of different tasks. T is the temperature, smoothing the weights of different tasks. If T is large then, then all the tasks have the same weight as 1.

Multi-task Training Curriculum (original) Due to the imbalanced sizes of the datasets for the three tasks, with the Quora dataset being significantly larger than the other two, training them sequentially could lead to the model forgetting earlier tasks. Conversely, training them in a multi-task manner—by calculating and combining each task's loss for backpropagation—typically requires the datasets to be of similar size; consequently, the performance on the second task could be compromised. Therefore, we propose a hybrid approach that alternates between sequential and simultaneous training to mitigate the drawbacks of each method. During the odd-numbered epochs, we perform sequential training, freezing all parameters except those specific to the current task, ensuring that each task does not influence the others and that the task with the larger dataset benefits. During the even-numbered epochs, we engage in simultaneous training, i.e., multi-task learning, fine-tuning all parameters while downsampling the second task to match the sizes of the other two datasets. We have found that this training curriculum can effectively improve performance.

Baseline For the baseline, I used the model architecture described above and the CLS pooling strategy. I simply add all loss together as the loss function, i.e., $L = L1 + L2 + L3$.

5 Experiments

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

Evaluation method Sentiment classification:classification accuracy; Paragraph detection:binary prediction accuracy; Semantic textual similarity: Pearson correlation.
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 204. Additionally, for training, all models utilize a learning rate of $1e-5$, a batch size of 32, and are trained for 20 epochs.

Results Firstly, I compare the mean pooling strategy with the baseline CLS pooling. The results are shown in Table 2.

	Overall	SST	Paraphrase	STS
Baseline(CLS)	0.667	**0.537**	0.647	0.636
Mean pooling	**0.713**	0.505	**0.731**	**0.805**

Table 2: Experiment result on dev set comparing CLS pooling and mean pooling

It shows that the overall score, as well as the scores for paraphrase and STS, have improved significantly, while the score for SST has dropped. I guess this is caused by the loss-imbalance problem, where the improvement in some tasks has hindered others. Therefore, I have tried two loss-balancing strategies described in the approach section, illustrated in Table 3.

	Overall	SST	Paraphrase	STS
Entity loss	**0.713**	0.505	**0.731**	**0.805**
Square Loss	0.708	0.512	0.720	0.783
Dynamic weight average	0.707	**0.523**	0.712	0.797

Table 3: Experiment result on dev set comparing loss-balancing techniques.

The results show that although the score for the SST task has improved, suggesting that the techniques help smooth out the conflict among tasks, the overall score has declined. This may be because loss imbalance is not the primary issue here, given that the state-of-the-art performance for the SST dataset hovers around 0.6, and reducing its score to 0.505 is not unacceptable. Instead, what likely impedes the overall performance is the quality of the embeddings, which motivates my further comparison of parameter adaptation methods.

The experimental results on the development set, which compare different parameter adaptation methods, are presented in Table 4. The results indicate that with the sequential training method, the first task performs poorly, while the second and third tasks show good performance. In contrast, with multi-task training, although the first and third tasks perform well, the second task exhibits poor performance. However, when employing my proposed hybrid training method, the performance of all three tasks has improved.

	SST	Paraphrase	STS
Sequential Training	0.305	0.810	0.803
Multi-task Training	0.505	0.731	0.803
Hybrid Training	**0.515**	**0.823**	**0.817**

Table 4: Experiment results on dev set comparing training curriculums

The experimental results on the development set, comparing different parameter adaptation methods, are shown in Table 5. They indicate that the combination of low-rank PAL and the top BERT layer method achieves the highest scores in the paraphrase and STS tasks. Upon uploading to Gradescope, its overall scores were recorded as 0.761 on the development set and 0.759 on the test set. The similarity of these scores suggests that this method possesses strong generalization capabilities. Among individual parameter adaptation methods, the top-performing method is the top layer adaptation, followed by the Houlsby adapter, then the low-rank method, and finally the PAL method. Overall, the method that performed best involves combining the low-rank and top

	SST	Paraphrase	STS
PAL	0.489	0.812	0.780
Low Rank	0.501	0.820	0.807
Proj.Atten.on Top	**0.511**	0.839	0.814
Houlsby Adapter	0.501	0.824	0.815
Low Rank + Top	0.509	**0.850**	**0.839**

Table 5: Experiment results on dev set comparing parameter adaption methods

layer adaptation methods, without employing any loss-imbalance tricks. It utilizes a hybrid training curriculum that alternates between sequential and simultaneous training strategies. The task-specific scores on the test set were 0.506 for SST, 0.856 for paraphrase detection, and 0.830 for STS.

6 Analysis

For the first two classification tasks, I further analyze their precision score, recall score and F1 score, as shown in Table 6.

	SST	Paraphase
Accuracy Score	0.509	0.856
Precision Score	0.538	**0.752**
Recall Score	0.462	**0.920**
F1 Score	0.459	0.828

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

It demonstrates that for the SST task, all scores are uniformly distributed, although they all fall below 0.6. This may be attributed to the task's increased difficulty, given its classification into five categories and the limited amount of data available. Moreover, for the second task of paraphrase detection, it is notably surprising to observe that the precision score is significantly lower than the recall score.

To delve deeper, I generated the confusion matrices for these two tasks as depicted in Figure 7 and Figure 8. In these matrices, each element M_{ij} epresents the number of samples with the true label i that were predicted as label j. Figure 7 reveals that for the SST multi-classification task, the diagonal line—which represents correct predictions—dominates. Following closely are the lines near and parallel to the diagonal, whereas the corner elements are nearly zero. This pattern indicates that 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 the second task, it is evident that the number of false positives significantly exceeds that of false negatives, suggesting the model is more prone to incorrectly predicting 0 as 1 rather than 1 as 0. Given more time, I would pursue further research to explore how to fine-tune the model to address this imbalance more effectively.

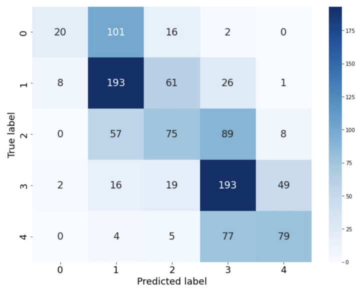

Figure 7: The confusion matrix for the SST task.

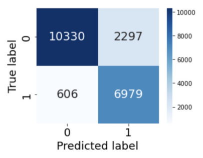

Figure 8: The confusion matrix for the paraphrase task.

For the third task, which is a regression task. I calculate more metrics and recorded in Table 7.

MSE	0.859
RMSE	0.927
MSLE	0.973
Median Absolute Error	0.575
Mean Absolute Error	0.719
Explained Variance Score	0.691
R2 Score	0.605

Table 7: More metrics for STS task.

7 Conclusion

In summary, throughout this project, I have explored various modifications to enhance model performance. Switching from CLS pooling to mean pooling proved to be quite effective. However, employing a loss-balance technique did not significantly enhance overall performance. I also proposed a hybrid multi-task training curriculum, which surpassed both the sequential training approach and the naive multi-task training method in effectiveness. Furthermore, upon implementing and comparing four parameter adaptation methods, I discovered that combining the low-rank and top attention method yielded the best results. Incorporating these findings, my multi-BERT model achieved scores of 0.761 on the development set and 0.759 on the test set, demonstrating that it does not suffer from overfitting. Additionally, by analyzing the precision and recall scores, I identified a significant imbalance between the number of false positives and false negatives in the paraphrase detection task, which will direct my future work. In conclusion, this project allowed me to explore various techniques for fine-tuning the multi-task BERT model and provided me with valuable practical experience.

References

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. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR.

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

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

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

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.