A Better Multitask BERT: Improving on Fine-Tuning

Stanford CS224N Default Project

Andrea Hurtado Sarah Teaw
Department of Computer Science
Stanford University
{andreahu, sarahlst}@stanford.edu

Abstract

The main goal of this project is to improve the original BERT model to perform sentiment analysis, paraphrase detection, and semantic textual similarity simultaneously. The first part of this project implements multi-head attention and transformer layers of BERT. The second improves this on downstream tasks by using cosine similarity fine-tuning for semantic textual similarity (STS), contrastive learning, and projected attention layers (PALs). Our results show that contrastive learning with cosine similarity fine-tuning for STS perform best for STS and has a higher accuracy and correlation overall, while PALs shows less of a deviation between each of the tasks from the baseline.

1 Key Information to include

- TA mentor: Jingwen Wu
- External collaborators (if no, indicate "No"): No
- External mentor (if no, indicate "No"): No
- Sharing project (if no, indicate "No"): No

2 Introduction

In the field of natural language processing (NLP), there is a strong demand for developing comprehensive and robust language models that closely mimic human language processing. Bidirectional Encoder Representations from Transformers (BERT) has made significant strides in creating robust language representations through pre-trainingDevlin et al. (2018). However, BERT exhibits considerable room for improvement on specific downstream tasks, a topic extensively studied in the literature. In this paper, we aim to enhance the accuracy and performance of minBERT, a minimalist version of BERT, across three tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. Existing methods typically focus on improving accuracy for individual tasks. This limitation arises because optimizations for one task can negatively impact the model's performance on others, due to differing task-specific requirements and objectives. Therefore, we investigate how to adjust BERT embeddings to achieve simultaneous improvements across all three tasks. In this study, we implement contrastive learning Gao et al. (2022), sentence BERT Reimers and Gurevych (2019), and projected attention layers to refine the BERT embeddings and boost performance overall. Our findings show that employing contrastive learning with cosine similarity fine-tuning yields superior performance for the Semantic Textual Similarity (STS) task, achieving higher accuracy and correlation metrics compared to other methods. However, the use of projected attention layers (PALs) in parallel with BERT demonstrates a more consistent performance across various tasks, exhibiting less deviation from the baseline results.

3 Related Work

Sentence BERT: Existing research have made great strides in improving BERT. Sentence-BERT (SBERT) focuses on sentence-pair regression tasks, which has the goal of predicting the relationship between two sentences. Sentence-pair regression tasks are applicable in determining semantic textual similarity (STS), where the goal is to determine how similar two sentences are in meaning, and paraphrase detection, which involves identifying whether two sentences express the same idea. The limitation of the original BERT model by Devlin et al. (2018) is that sentence embeddings are not independently computed, therefore causing the model to be extremely computationally expensive on sentence-pair regression tasks Reimers and Gurevych (2019). Therefore, as the authors of SBERT found, deriving sentence embeddings can reduce the computational expense and improve performance accuracy on sentence-pair regression tasks.

Projected Attention Layers: Beyond just sentence-pair regression tasks, in order for the model to perform well on all three tasks defined in the introduction, multitask learning is also essential to ensure the model can generalize and adapt to different objectives simultaneously. Caruana (1997) finds multitask learning can improve the learning of multiple tasks by training on each in parallel and using shared hidden states amongst the tasks. The hard-parameter sharing implementation, where all tasks have the same hidden layers and specific output layers, is deemed more reasonable by Stickland and Murray (2019) to keep the numbers of parameters low.

Contrastive Learning: Contrastive learning is a technique that has been used in the context of machine learning that maps similar representations closer and less similar representations farther. Chen et al. (2020) found in the field of vision that contrastive learning works best with large batch sizes and greater training steps. The study also found that linear classifiers applied after contrastive learning improved in accuracy, giving promise to its applications in natural language processing.

4 Approach

For the first part of this project, the multi-head attention and transformer architecture of BERT is borrowed and implemented for the sentiment analysis task. The second part improves on the existing model by fine-tuning on the three downstream tasks using projected attention layers, contrastive learning, and sentence BERT.

4.1 Models and Techniques

To improve on the BERT model, fine-tuning on multiple tasks simultaneously is done by adding the losses of each task. Our first improvement is to semantic textual similarity. As Reimers and Gurevych (2019) found, the STS of two embeddings can be improved using siamese networks under the Sentence-BERT (SBERT) model. Because averaging the BERT output layer were found by the study to output poor sentence embeddings, SBERT modifies the original BERT model by implementing pooling to get sentence embeddings. Our model uses the regression objective function from the Reimers and Gurevych (2019) study, which entails calculating the cosine similarity between the 2 pooled sentence embeddings, and using mean squared error for the loss function. A visual of the similarity score calculation is shown in Figure 1. From the similarity score, we add 1.0 to obtain a positive scale and multiply by 5.0/2.0 to match the labels between 0-5 in the SemEval STS.

Furthermore, our model improves sentence embeddings by implementing the simple contrastive sentence embedding framework (SimCSE), as studied by Gao et al. (2022). Unsupervised contrastive learning is implemented, which passes the same sentence through the BERT model twice, using independently sampled dropout masks in Transformers. The sentence embeddings are then treated as positive pairs and the original SimCSE loss function is implemented, shown in Equation 1.

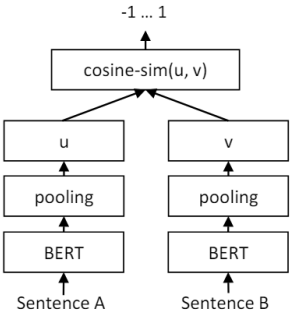

Figure 1: SBERT architecture to compute similarity scores. Figure comes from original paper.

$$\ell_i = -\log \frac{e^{\text{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z_i'})/\tau}}{\sum_{j=1}^{N} e^{\text{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_j^{z_j'})/\tau}}$$

Equation 1: Training objective loss function for unsupervised SimCSE.

In this equation, $\mathbf{h}_i$ and $\mathbf{h}_j$ are BERT encoded sentence embeddings, z' and z are distinct dropout masks from the Transformer, and sim() represents the cosine similarity function. In our implementation, we follow the best temperature hyperparameter found by Gao et al. (2022) on STS-B, which was 0.05. Our model implements contrastive learning prior to the multitask training.

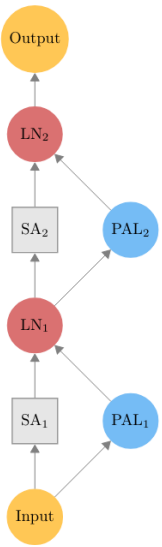

Figure 2: PALs architecture to compute similarity scores. Figure comes from original paper.

As we read over the Stickland and Murray (2019) paper, we found Projected Attention Layers (PALs) technique could help streamline the task-specific attention with the parallel BERT layers. Instead of modifying the multi-task itself, we added parameters within BERT, where the self.pals functionality allowed for the task-specific function to take place, shown in Equation 2.

$$TS(\underline{\mathrm{h}}) = V^D g(V^E \underline{\mathrm{H}})$$

Equation 2: Task-specific functions for PALs.

The projection layer shared among BERT is V^E and V^D. Following the recommendation of the source code and paper, we had the V^E and V^D follow the form of $d_s \times d_m$, where the values defaulted to $d_s = 204$ and $d_m = 768$. As a parallel to BERT, we observed the architecture follow closely to Figure 2.

Loading up to train, we experimented with 'anneal sampling,' as mentioned in the PALs paper, but we observed that our accuracy results remained best when we trained on round robin. In the use of anneal sampling, as demonstrated in Equation 3, α changes with each epoch e, where we also included the suggested 'arbitrary epoch' of training steps in the paper: 2400 training steps.

$$\alpha = 1 - 0.8\frac{e-1}{E-1}$$

Equation 3: Developed 'anneal sampling' equation for training based on PALs paper.

4.2 Baselines

The baseline of our model for sentiment analysis is minBERT we implemented based on the original BERT model by Devlin et al. (2018), which has the following accuracies:

- Fine-tuning last linear layer for SST: 0.409
- Fine-tuning last linear layer for CFIMDB: 0.788
- Fine-tuning full model for SST: 0.513
- Fine-tuning full model for CFIMDB: 0.971

The baseline for our multitask BERT model is the minBERT model with simple classification heads for each of the three tasks. Sentence pair embeddings were simply passed through the BERT layer and dropout layer, then concatenated together as inputs to the classifiers. The dropout probability is 0.1 across all embeddings. Cross entropy loss, binary cross entropy, and mean square error loss were used for sentiment analysis, paraphrase detection, and semantic textual similarity respectively.
8,544 examples of the Stanford Semantics Treebank dataset, 6,040 examples of the SemEval STS dataset, and 10,000 examples of the Quora were used for the baseline. By setting up training to perform round robin over all three tasks, each example in the datasets accounted for in the baseline. We trained on 5 epochs with default parameters: batch size of 8, learning rate of 1e-5, Adam optimizer, and last linear layer fine-tune mode.

5 Experiments

5.1 Data

In sentiment analysis, Stanford Sentiment Treebank[1] (SST) parsed with the Stanford Parser[2]and the Complex Features IMDb dataset[3] (CFIMDB) datasets are used. The Stanford Sentiment Treebank (SST) consists of sentences from movie reviews in which each phrase has a label of negative, somewhat negative, neutral, somewhat positive, or positive, with discrete numerical values 0-5 respectively. The SST dataset is split into train | dev | test, using the following:

- train (8,544 examples)
- dev (1,101 examples)

[1] https://nlp.stanford.edu/sentiment/treebank.html
[2] https://nlp.stanford.edu/software/lex-parser.shtml
[3] https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

- test (2,210 examples)

In paraphrase detection accuracy, the Quora Dataset[4] is used to train and test our BERT model. It contains question pairs with labels indicating whether particular instances are paraphrases of one another, with binary labels of 0 or 1. Due to its large size, we choose the first 10,000 examples to train on.
The dataset is split into train | dev | test, using the following:

- train (283,010 examples)
- dev (40,429 examples)
- test (80,859 examples)

In semantic textual similarity, the SemEval STS Benchmark Dataset[5] is used, which consists of sentence pairs with varying similarity on a scale from 0 (unrelated) to 5 (equivalent meaning). The dataset will be useful to see the correlation of the true similarity values against the predicted similarity values. The dataset is split into train | dev | test, using the following:

- train (6,040 examples)
- dev (863 examples)
- test (1,725 examples)

The inputs for the sentiment and sentence regression tasks model consist of two primary components. The first component, input IDs, represents the tokenized text input where each token is mapped to an index corresponding to tokens in the BERT vocabulary. The second component, attention masks, comprises binary masks that indicate word presence, with a value of 0 denoting padding and a value of 1 indicating real tokens. The outputs of the sentiment and sentence regression tasks model include two key elements. The first element, logits, provides the probabilities for each class. The second element, predicted labels, consists of the final class predictions. For contrastive learning, following the dataset used in training unsupervised SimCSE by Gao et al. (2022), the English Wikipedia dataset wiki1m_for_simcse[6] dataset is used. The dataset contains unlabeled sentences. Due to its large size (1,000,000), we choose the first 25,000 examples for training. The sentence input IDs and attention masks follow a similar form as the other data sets.

5.2 Evaluation method

The evaluation metric used is the mean reference accuracies over 10 random seeds, with specified standard deviation. In performing sentiment analysis and paraphrase detection on SST and QQP respectively, accuracy is used for evaluation. In evaluating semantic textual similarity on the SemEval STS Benchmark dataset, Pearson correlationAgirre et al. (2013) between true similarity values and predicted similarity values are used. This is because the values are continuous rather than discrete. The average of these three values are also used for overall evaluation.

5.3 Experimental details

Experiments were ran using the minBERT model, which had hidden size of 768. The multitask model is trained on 10 epochs with 2400 training epoch steps with default parameters: batch size of 8, learning rate of 1e-5, Adam optimizer, and last linear layer fine-tune mode. For each task, a dropout layer is applied before the linear layer with a dropout probability of 0.3.

5.4 Training Approaches

A challenge of maximizing performance on different datasets is accounting for the differing lengths. Due to the QQP dataset being considerably larger in size than SST and STS, we experimented with training sequentially, cycling over max-size (QQP in this case), round robin, and annealed sampling. Due to time constraints, one epoch was run for each training approach with cosine similarity extension

[4]https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

[5]https://aclanthology.org/S13-1004.pdf

[6]https://huggingface.co/datasets/princeton-nlp/datasets-for-simcse/blob/main/wiki1m$_{f}or_{s}imcse.txt$

implemented and found round robin produced results the fastest and thus used this approach for further training. In round robin, during each epoch, the training loop will grab a batch from each downstream tasks' dataset. It checks for which dataset has the most examples and consequently runs all other proportional to the longest length, starting new batch runs for shorter datasets. Given more time and computing capabilities, further research can be done on different training approaches.

5.5 Results

		SST Accuracy	PARA Accuracy	STS Correlation	Overall Score
DEV	Baseline (CLS pooling for all 3 downstream tasks)	0.493	0.723	0.345	0.520
	Cosine Similarity (on STS)	**0.510**	**0.733**	0.438	0.560
	Cosine Similarity + SimCSE	**0.510**	0.722	**0.481**	**0.657**
	Cosine Similarity + PALs	0.508	0.728	0.414	0.648
	Cosine Similarity + SimCSE + PALs	0.490	0.724	0.445	0.553
TEST	Cosine Similarity + SimCSE	0.527	0.725	0.468	0.662

Table 1: Experiment performance on TEST and DEV sets of different model architectures, accuracies and Pearson correlations reported.

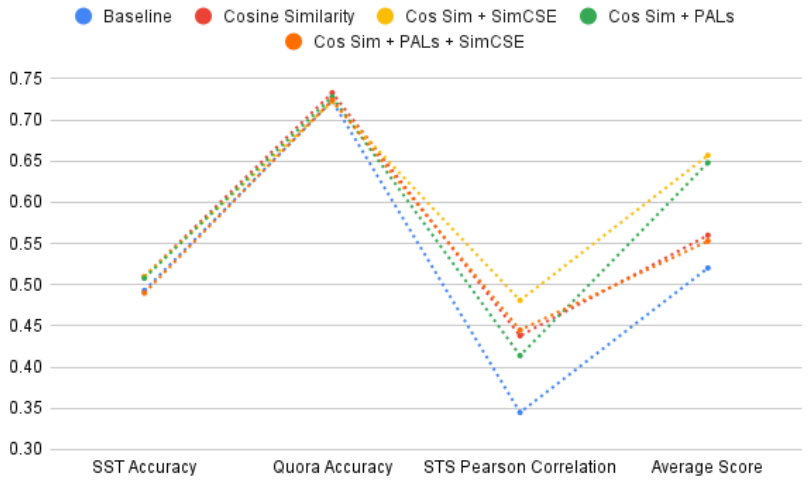

Figure 3: Visualization of experiment performance on TEST and DEV sets of different model architectures, accuracies and Pearson correlations reported.

The best test accuracies and Pearson correlation scores we obtained on the test leaderboard are reported in Table 1, with SST 0.527, QQP 0.725, STS 0.468, and overall 0.662. Between the different layers of fine-tuning and model adaptations, cosine similarity implemented only on the STS classifier and not on the QQP classifier was found to perform better than cosine similarity on the QQP classifier, therefore we reported these scores in Row 2. We found that simply concatenating the QQP sentence pair embeddings before passing them into the paraphrase classifier linear layer had the best accuracy, therfore we used this in subsequent training runs. Overall, using round-robin sampling over with cosine similarity on STS and unsupervised SimCSE performed the best, with

overall accuracy of 0.657. Surprisingly, PALs implemented in conjunction with SimCSE did not improve overall accuracies, as SimCSE only slightly improved STS Correlation.

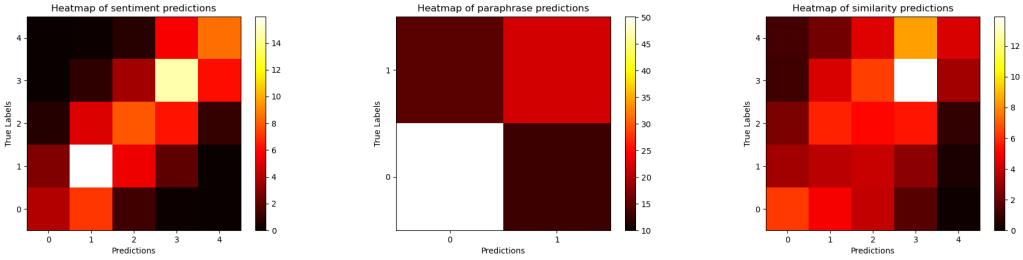

Figure 4: Heatmap of predicted labels vs. ground truth labels for sentiment, paraphrase, and similarity respectively.

The heatmaps shown in Figure 4 plots the predicted labels against the ground truth labels for each task: sentiment analysis, paraphrase detection, and semantic textual similarity. This heatmap was created using the data from implementing unsupervised SimCSE and cosine similarity on STS, the highest average accuracy we obtained. The intense colors (closer to white) represent a greater percentage of predictions and duller colors (closer to black) represent a lesser percentage of predictions. As seen clearly in sentiment analysis, a diagonal pattern of highest intensity translates to a higher accuracy, which is expected as the SST accuracy reported in Table 1 Row 3 is 0.510. While paraphrase detection is difficult to visualize through a heatmap due to the binary labels, we see that there is a roughly diagonal pattern and our model is best at predicting when sentences are not paraphrases of each other. Finally, there is more spread but also higher intensity of the diagonal pattern in STS, showing our model may be off by a larger range but still in the general vicinity of similar sentiment.

6 Analysis

A qualitative evaluation of our model reveals several insights into implementing several minBERT improvements. The improved performance of the STS classifier with cosine similarity suggests that this approach does well at capturing the semantic similarity between sentence pairs. Conversely, the lack of improvement when applying cosine similarity to the QQP classifier indicates that the QQP task might require alternative fine-tuning methods to achieve better results, limiting the applications of cosine similarity in binary classification. The results show that paraphrase detection tasks have shown best performance when using concatenated BERT embeddings alongside a simple classification head. This suggests that cosine similarity may not be most applicable to all sentence pair regression tasks.

The highest overall score being achieved by unsupervised SimCSE and cosine similarity is unexpected, but its performance on STS is as expected. As Gao et al. (2022) demonstrated, simple contrastive learning methods work surprisingly well for tasks involving semantic similarity. Our findings closely align with theirs, highlighting how these methods can enhance the model's representation of semantic relationships between sentences. This connection between what tasks require and how we fine-tune models highlights the importance of adjusting our approach to suit each task specifically.

In contrast, Parameterized Attention Layers (PALs), which modulate attention mechanisms, may not offer the same level of semantic alignment without further task-specific adaptation. While PALs have shown promise in improving model interpretability and performance in certain contexts as shown by Misra et al. (2020), its performance can vary depending on the task requirements and model architecture. Therefore, integrating PALs into models like SimCSE may require further research to reduce conflict between with the target task's objectives.

Projected attention layers were a as a promising addition to improving the BERT model performance across multiple tasks. PALs have been found to simultaneously improve accuracies in various downstream tasks, with notable impacts on tasks such as SST accuracy and STS correlation. In comparison to the baseline, PALs improved the model by a significant amount, with an overall +0.128 rise in accuracy and correlation. However, it did not out-perform SimCSE, which leads us to believe it requires further exploration and adjustment.

Furthermore, the utilization of round-robin sampling has been shown to achieve the best training runtime.

7 Conclusion

Through the implementation of cosine-similarity on STS, unsupervised contrastive learning, and projected attention layers on the multitask minBERT model, the results show the success of cosine similarity fine-tuning and contrastive loss for STS. The varied results with PALs suggest PALs and contrastive learning do not function well together in the current model but show direction for future research to improve overall model performance. Different accuracies obtained from our exploration on training approaches also show that our current model may be limited by the round-robin approach and may be able to perform better with different sampling methods. Limitations of our work was time and computing capabilites, however, in the future, exploring supervised contrastive learning, annealed sampling, and other multitask improvements holds promise for further improving the robustness and efficiency of our approach.

8 Ethics Statement

As we developed our minBERT, we considered the societal biases concerned with language models. Addressed by Stanford's Ethics and Society Review (ESR) Bernstein et al. (2021), the consequences of language models that express gender or age bias can translate to unintended ripple effects such as underpaid employees or limited opportunities. While training minBERT, training on data that enforces individual fairness is essential. Ideally, that means adding more datasets that stress group fairness; however, we also recognize that the default project explicitly states that we may only use the given datasets to train | dev | test. Thus, for the sake of the project, we will not add additional datasets, but we advocate for trained datasets that reflect greater representation. Additionally, since our BERT will be exclusively adapted to the given dataset, the sentiment analysis will likely fall short of capturing the sentiment variety of different cultures. That is, the given Stanford Sentiment Treebank dataset Socher et al. (2013) exclusively used www.rottentomatoes.com reviews, meaning that our trained data for sentiment analysis only expresses sentiments shared by those who use that site. We recognize that having a select website for movie reviews creates a sample size of selective sentiments, possibly not capturing sentiments varied among different cultures. Thus, we would also advocate for more datasets that represent different cultures and their form of expression to avoid a biased result in the form of sentiment.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Michael S. Bernstein, Margaret Levi, David Magnus, Betsy Rajala, Debra Satz, and Charla Waeiss. 2021. Esr: Ethics and society review of artificial intelligence research.

Rich Caruana. 1997. Multitask learning. *Mach. Learn.*, 28(1):41–75.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

Ishan Misra, Martial Hebert, and C. Lawrence Zitnick. 2020. Shuffle and learn: Unsupervised learning using temporal order verification. In *European Conference on Computer Vision*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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

Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning.