minBERT and Downstream Tasks

Stanford CS224N Default Project

Ziyang Ding
Department of Statistics
Stanford University
zd26@stanford.edu

Daniel Zou
Department of Computer Science
Stanford University
dlzou@stanford.edu

Abstract

We implement a minimal version of the BERT transformer language model and foster understanding of how to fine-tune such a model specifically for downstream tasks. Given the pre-trained model weights of bert-base-uncased as the baseline, we develop extensions in order to build a single model that can simultaneously perform inference on three separate NLP tasks, namely sentiment analysis, paraphrase detection, and semantic textual similarity. We have extended the baseline model with task-specific last layers, and we investigate the effect of training methodology, model architectures, and loss functions on prediction accuracies across all three tasks. Our experiments reveal the outsized impact of a single architectural change and highlights the versatility of the BERT pretrained model itself, while prompting further questions surrounding model explainability.

1 Key Information to include

Mentor: Johnny Chang External Collaborators (if you have any): n/a Sharing project: n/a

Ziyang implemented training and loss functions for fine-tuning task extensions. Daniel implemented first versions of BERT model and fine-tuning. Both contributed equally to experiments and the final report.

2 Introduction

Natural Language Processing (NLP) is at the cutting edge of empowering machines to understand, interpret, and respond to human language in meaningful ways. Among the various tasks tackled by NLP, sentence-level activities such as sentiment analysis, paraphrase detection, and semantic textual similarity play crucial roles in numerous applications, ranging from automated customer service to content analysis and beyond. However, the complexity of human language, with its implicit meanings, subtleties, and context dependencies, makes these tasks particularly difficult.

Historically, the predominant approach in natural language processing (NLP) research was to invent algorithms that targeted individual tasks. For example, early attempts at sentiment analysis, such as Tan et al. (2015), were in the form of rule-based systems that rely on manual, bespoke procedures like aggregating lists of polarizing words and building heuristics. Later, researchers turned to machine learning as a more powerful technique for classification, building and tuning model pipelines that were specialized for the single task. The disadvantage of this approach is that the theory and expertise gained from improving a solution to one NLP task seldom benefits solutions to other task.

The introduction of models like GPT (Generative Pre-Trained Transformers, Radford and Narasimhan (2018)) and BERT (Bidirectional Encoder Representations from Transformers, Devlin et al. (2019)) has markedly advanced the capabilities of the field. In particular, these works popularized the practice of first *pre-training* a model on a large, diverse text corpus in order to build general language understanding, then *fine-tune* model parameters to specialize in a downstream task, possibly with the

addition of a small last layer. This approach has yielded ground-breaking performance improvements compared to past methods.

Despite BERT's strong foundation for addressing a variety of NLP tasks, the conventional fine-tuning process is designed for a single downstream task, and does not consider the additional complexity brought by training one model on multiple tasks. Doing so introduces the problem of task interference, where learning from one task adversely affects another, resulting in difficulty in effectively balancing multiple objectives within a single model framework.

This project studies the problem of fine-tuning a single pre-trained BERT model to multiple tasks. We examine the impact of

1. Change of a loss function in sub-tasks
2. Change of a model architecture in paired input sub-tasks
3. Change of fine-tuning procedure

on the overall performance of the entire combined model on all its tasks. More details will be discussed in Section 4

3 Related Work

The authors of the GPT paper Radford and Narasimhan (2018) pre-trained a transformer decoder model on a large unlabeled corpus with diverse forms of text, using the generative task of next word prediction. Afterward, the model can be fine-tuned on smaller datasets for specific tasks like textual entailment and question answering, exceeding previous best accuracies in most of such tasks. This work overcomes the hurdle of lack of labeled data for specific NLP tasks, and popularizes the approach of task-agnostic pretraining.

BERT Devlin et al. (2019) builds upon previous work by introducing a bidirectional transformer encoder model that is more powerful for creating sentence embeddings. The model is pre-trained using the masked language modeling and the next sentence prediction tasks. The BERT authors found that they could achieve state-of-the-art performance in a wide range of NLP tasks just by adding a single output layer to BERT and fine-tuning.

We further researched related work in the following three subcategories.

3.1 Difference in STS loss function

Various loss functions have been explored for fine-tuning BERT models on semantic textual similarity (STS) tasks. Huang et al. (2024) proposes a novel Consistent SENTence embedding (CoSENT) framework that optimizes a Siamese BERT network using a supervised objective function exploiting ranked similarity labels of sample pairs, with a uniform cosine similarity-based loss for training and prediction. Similarly, CS224N et al. (2023) investigates a multitask approach, combining sentiment analysis, paraphrase detection, and STS, using a weighted multitask loss function and similarity task fine-tuning.

Yasui et al. (2019) explores using semantic similarity as a reward for reinforcement learning in sentence generation, employing a BERT-based scorer fine-tuned on the STS task to estimate similarity scores. Additionally, Ma et al. (2023) presents a comprehensive study of the miniBERT model for STS, sentiment analysis, and paraphrase detection, introducing extensions like gradient surgery, cosine similarity fine-tuning, and sequential learning.

In another study, Zhang et al. (2021) proposes an Attention-based Overall Enhance Network (ABOEN) for Chinese STS, utilizing convolutional neural networks with soft attention layers and a channel attention mechanism to capture interactive features between sentences. These studies collectively illustrate the diverse approaches and enhancements applied to BERT models for improving performance on STS and related tasks.

3.2 Parallel or concatenated use of BERT

Several studies have explored the efficacy of different BERT configurations for NLP tasks involving sentence comparisons, such as paraphrase detection and semantic textual similarity (STS). The

Sentence-BERT (SBERT) approach uses a Siamese BERT network where each sentence is passed through a separate BERT model, and their embeddings are combined to compute similarity. This method significantly improves performance and efficiency over the traditional BERT model by using cosine similarity between sentence embeddings, thereby reducing computational overhead and improving embedding quality Reimers and Gurevych (2019).

Another study comparing BERT and ALBERT models found that while concatenating sentences and passing them through a single BERT model works well for tasks requiring direct sentence interaction, using separate BERT models for each sentence and then combining their embeddings is more effective for capturing individual sentence nuances and reducing interference Lan et al. (2019).

The ColBERT study also supports the use of separate BERT models, demonstrating that this approach offers better scalability and flexibility in capturing sentence-level nuances compared to the concatenated approach, which may struggle with maintaining the integrity of individual sentence embeddings Khattab and Zaharia (2020). These findings collectively suggest that using separate BERT models for each sentence, followed by combining their embeddings, generally yields better performance for sentence comparison tasks.

3.3 Finetuning full model or last layer

Research has investigated various strategies for fine-tuning BERT models for multitask learning, including whether to fine-tune the entire model, just the last layer, or a combination of both. One study found that while fine-tuning the entire model generally provides better performance, it also increases computational cost and potential instability. Fine-tuning just the last layer offers faster training and stability but may result in a performance drop. A hybrid approach, fine-tuning the last few layers, is suggested to balance performance and efficiency et al. (2021a).

Another study proposes a partial fine-tuning method where only the top layers of BERT are fine-tuned while keeping the rest of the model frozen. This approach reduces computational overhead and prevents task interference, achieving nearly the same performance as full fine-tuning while significantly reducing resource requirements. This method is especially beneficial for iterative and incremental task development Wei et al. (2022).

Further research on domain-specific tasks in the biomedical field compared full model fine-tuning, last layer fine-tuning, and sub-domain adaptation. It concluded that while full fine-tuning delivers the best task-specific performance, combining sub-domain adaptation with selective layer fine-tuning enhances stability and efficiency. This hybrid approach allows the model to retain general knowledge while effectively adapting to specific tasks et al. (2021b).

4 Approach

4.1 Implementing minBERT Baseline

We constructed the base minBERT model by referencing several foundational works in large language model research literature. First, we implemented the self-attention module using scaled dot product attention, which was proposed by Vaswani et al. (2023) and defined as

$$\text{Attention}_i(\mathbf{h}_j) = \sum_t \text{softmax}\left(\frac{W_i^q \mathbf{h}_j \cdot W_i^k \mathbf{h}_t}{\sqrt{\frac{d}{n}}}\right) W_i^v \mathbf{h}_t$$

Our self-attention module uses multi-head attention by concatenating n attention head hidden states, and it performs masking to zero out elements where the source token is padding.

The transformer block forward pass starts with a multi-head self-attention layer, then performs layer normalization Ba et al. (2016) on the attention layer residuals, with dropout enabled during training. The normalized outputs are put through a standard feed-forward network using the GeLU Hendrycks and Gimpel (2023) non-linear activation function, and then a final layer normalization is applied to get the transformer block output.

We implement the AdamW optimizer Kingma and Ba (2017) for efficient training. AdamW is a an algorithm for gradient-based stochastic objective functions that adapts the learning rate of each

parameter to estimates of lower-order moments, providing better convergence characteristics during our fine-tuning process.

4.2 Fine-Tuning

To adapt the hidden state that is produced by the baseline BERT model to our downstream classification tasks, for each task, we append unique last layers to the end of the model that project the first embedding token into the task output space, whose values are treated as logits. During inference, the output space for each task is:

1. Sentiment analysis: five discrete classes ranging from 0 to 4, representing negative to positive
2. Paraphrase detection: binary labels of 0 or 1
3. Semantic textual similarity: a continuous range from 0 to 5, with 4 being most similar

For the baseline, a different loss function is used for each task depending on the appropriate learning objective. Standard cross entropy is used for sentiment classification. The paraphrase detection task has binary output, so binary cross entropy is used. Since the semantic textual similarity task is evaluated based on Pearson correlation, we use correlation as the objective, given by

$$\rho = \frac{\text{Cov}(x, y)}{\sqrt{\text{Var}(x)\text{Var}(y)}}$$

4.3 Model adjustments

As informed by prior work, we designed experiments to how each of the following three factors affects combined task prediction accuracy:

1. Change of a loss function in sub-tasks
2. Change of a model architecture in paired input sub-tasks
3. Change of fine-tuning procedure

4.3.1 Change of loss function in STS

For item 1, We focus on changing the loss function for semantic textual similarity. We are interested in testing how mean square error loss, defined below, compares to our baseline correlation objective:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

We will compare the testing result with the our proposed pearson correlation coefficient loss defined in the previous section.

4.3.2 Sub task model architecture

Apart from sentiment analysis, the 2 other sub tasks, including paraphrase detection and semantic textual similarity, all involves comparison between 2 sentences as input. Hence, the question of should we separately embed each sentences into a embedding using minBERT, then take the 2 embedding as input for the downstream classification/prediction task, or should direct concatenate the tokens of both sentences into a longer sentence, and feed the concatenated sentence as 1 input variable into minBERT as a single embedding, and run the downstream task. A figurative illustration of the 2 approach is shown below in Figure 1.

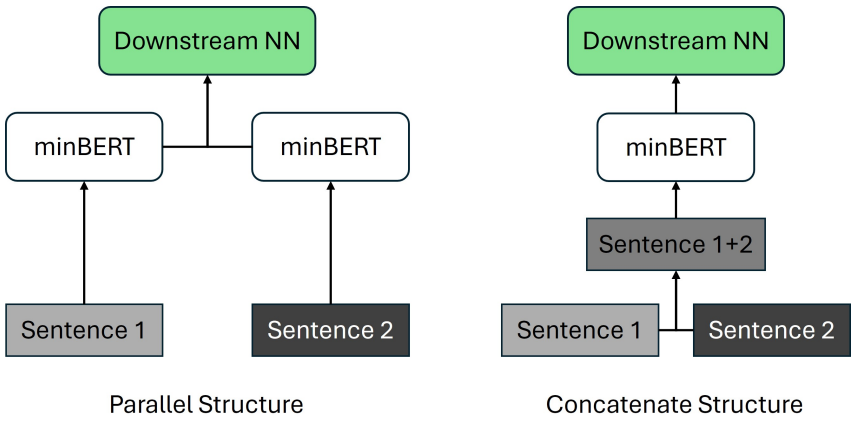

Figure 1: Parallel vs Concatenated subtask structure

4.3.3 Fine tune scheme

To control variable, we force all training scheme to have 5 epochs in total. We explore 5 epochs on all models with 1) full model fintetuning for all 5 epochs, 2) full model finetuning for the first epoch, then stop the gradient for the minBERT and only allow last layer finetune for the rest 4 epochs, and 3) all 5 epochs finetuning the last layer.

5 Experiments

5.1 Data

We use the datasets that have been provided for the default project and that specifically target the three downstream tasks.

For sentiment analysis, we use the Stanford Sentiment Treebank (SST) Socher et al. (2013) dataset. The dataset consists of over 10,000 sentences extracted from movie reviews and parsed using the Stanford parser into over 200,000 unique phrases, which are annotated by humans with five different sentiment levels ranging from negative (0) to positive (4).

For paraphrase detection, we use the Quora dataset that contains over 400,000 question pairs that have binary labels indicating whether each pair constitutes paraphrases of one another.

For semantic textual similarity, we use the SemEval STS Benchmark Dataset Agirre et al. (2013). It has over 8,000 sentence pairs that have been labeled of a scaled from 0 (unrelated) to 5 (equivalent meaning).

5.2 Evaluation method

We utilize the standard evaluation metric included for the 3 extension tasks. We uses accuracy for sentiment prediction task, accuracy for paraphrase detection classification task, and the Pearson correlation coefficient for the semantic textual similarity. During the process of training, we also monitor f1 and f2 score for both sentiment prediction and paraphrase detection to prevent skewed data impact, but the final evaluation is still based on accuracy.

5.3 Experimental details

We use the BERT Base variant for all experiments, which has 12 transformer blocks and a hidden state size of 768. Due to the long time required for fine-tuning all model parameters, we fine-tune all models for 5 epochs, with a learning rate parameter of 1e-05, and dropout probability of p=0.3.

We trained 12 models in total to cover all combinations of loss functions, model architectures, and training procedures described in our approach 4.

STS Objective	Training Epochs	Architectures					
		Parallel			Concat		
		SST	PARA	STS	SST	PARA	STS
Pearson	5 last layer	0.459	0.711	0.083	0.453	0.767	0.556
Pearson	1 full + 4 last layer	0.388	0.785	0.361	0.466	0.891	0.871
Pearson	5 full	0.494	0.811	0.337	0.490	0.891	0.868
MSE	5 last layer	0.326	0.701	0.138	0.324	0.737	0.430
MSE	1 full + 4 last layer	0.421	0.774	0.235	0.420	0.801	0.729
MSE	5 full	0.494	0.791	0.363	0.478	0.887	0.875

Table 1: Comparing DEV scores for all experiments

SST	PARA	STS	Overall
0.486	0.892	0.873	0.772

Table 2: Best TEST score for final submission (Pearson, 5 full, Concat)

5.4 Results

Table 1 shows the results for all 12 models we trained when evaluated on the DEV dataset. SST represents the sentiment analysis task, with values representing accuracy; PARA represents the paraphrase detection task, with values representing accuracy; STS represents the semantic textual similarity task, with values representing correlation. In the STS Objective column, Pearson represents the loss fucntion based on Pearson correlation coefficient, and MSE represents mean squared error loss. The two types of architectures we trained for the PARA and STS tasks are denoted as Parallel and Concat.

When analyzing Table 1, we observe that for the PARA and STS tasks, the Concat architecture yields significantly better performance. The PARA task sees approximately 10% improvement in accuracy, while STS correlation is dramatically improved from about 0.35 to about 0.87 for the best trained models. This is a surprising result, as we had expected good performance for the Parallel architecture which is based on the Sentence-BERT approach.

Comparing the three training schemes, we find that there generally is performance improvement as the model is fully trained for more epochs, which aligns with our prediction. For some instances, like the architecture models trained on the Pearson objective, training the full model for only one epoch plus 4 short epochs of training the last layer yielded results nearly as good as training fully for five epochs. However, this is not the case when models were trained on the MSE objective. This suggests that Pearson objective could be more conducive toward optimization given our evaluation metric, but that MSE loss still converges to similar performance given more training.

Our final submission, shown in Table 2, uses the Concat architecture for the latter two pair-wise tasks, and the full model's parameters has been fine-tuned for 5 epochs, with the Pearson objective being used for STS.

6 Analysis

Our qualitative analysis boils down to the following observations:

PEARSON CORRELATION COEFFICIENT AS A LOSS

Usually the pearson correlation coefficient is massively criticized as a loss function due to its sensitivity to outliers, non-differentiability, and invariance to scale. It focuses on the direction rather than the magnitude of differences and assumes linear relationships, which may not suit all data. These factors can lead to inefficient learning and suboptimal model performance. However, we do observe that the performance of using simple correlation coefficient can actually meet performance of MSE.

We guess that the reason of this could be, since the training not only focuses on maximizing the correlation, but also includes other tasks such as SST and PARA. These other tasks provided perturbations on the model weights, therefore prevented degenerative gradient from forming, hence brought robustness to this instable loss function.

PROBLEM OF PARALLELING BERT FOR SEPARATE EMBEDDING

We noticed that when introducing paralleling BERT to obtain separate embedding for 2 sentences will introduce a big loss on its performance. This seems to be counter intuitive at first, since the way human compare 2 sentences is to form independent opinions and understandings on them first, and based on the understanding any followup suggestions on similarity, paraphrase etc, are made. We believe, if we were to proceed in improving the extension (last layer) from simple linear output to more sophisticated output NN model, the performance of parallelized embedding approach could also work well.

But to qualitatively explain the performance boost in using concatenated embedding, one possible impression is that the concatenated input is somewhat similar to a "joint variable", in which interactions and relationships between the 2 sentences are also measured and considered during the embedding phase. Though less computing power is used, the "independence" structure between the 2 sentences are not pre-assumed (just like the cross covariance matrix, if both sentence are considered in a multidimensional random variable setting). Hence we are observing a improvement on the concatenated joint embedding.

PROBLEM OF OVERFITTING DURING FULL MODEL FINETUNING

Since finetuning the entire model will give more flexibility to the model to better adapt to the data, we usually anticipate full model finetuning to achieve better performance than last layer finetuning. However, there could still be exceptions. By comparing results in Table 1, the Pearson Concat class, 1 full + 4 last layer, with that of 5 full model, we do see a slight decrease in performance in STS task, though the performance on SST is improved, and performance on PARA is on par. This, to some extends, tells that aggressive finetuning on the full model could potentially mess-up the minBERT weights, thus causing overfitting the model and lead to worse predictions.

Switching from separate forward passes of paired inputs to concatenated forward pass resulted in the largest performance improvement. This suggests that the BERT architecture is highly generalizable.

7 Conclusion

In this project, we fine-tune a pre-trained BERT model on three NLP tasks—sentiment analysis, paraphrase detection, and semantic similarity classification—with the goal of achieving the highest combined performance on corresponding evaluation metrics. We first implement the baseline model with multi-head self-attention, the AdamW optimizer, and forward passes for each of the three tasks. Then, we investigate how loss functions, model architecture, and fine-tuning procedure affect performance.

A key finding from our experiments is that for the paired input tasks of paraphrase detection and semantic textual analysis, the Concat architecture performs signification better than the Parallel architecture. We also observed that the Pearson objective was more effective than the MSE objective when it comes to fine-tuning for the STS task. We confirmed our hypothesis that fine-tuning the full model usually leads to better performance than fine-tuning the last layer, though we also found some indications of overfitting.

Time and access to compute were limitations during the course of our project. We believe that given more resources, we may find more interesting and conclusive results when it comes to our comparison of loss functions.

8 Ethics Statement

In online social media platforms, NLP models are widely deployed to analyze and extract information about users. However, many users may be unaware that their words are being used for such purposes.

Furthermore, there may be vulnerable groups like children from whom it would be both legally and ethically unacceptable to extract information in any situation. It is therefore crucial that NLP models are deployed with care, and that users are consenting.

The recent paradigm of pre-training massive language models is extremely energy intensive, so there is potential for indirect harm to the environment if energy sources are not choosen carefully. All institutions in the business of pre-training large models should be conscious of their energy footprint and invest in renewable energy sources as well as more efficient training methods.

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.

Authors et al. 2021a. A closer look at how fine-tuning changes bert. *arXiv preprint arXiv:2106.14282*.

Authors et al. 2021b. Investigation of improving the pre-training and fine-tuning of bert model for biomedical relation extraction. *BMC Bioinformatics*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Stanford CS224N, Default Project, Prarthna Khemka, and Grace Casarez. 2023. Bert with multitask fine-tuning and loss construction.

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

Dan Hendrycks and Kevin Gimpel. 2023. Gaussian error linear units (gelus).

Xiang Huang, Hao Peng, Dongcheng Zou, Zhiwei Liu, Jianxin Li, Kay Liu, Jia Wu, Jianlin Su, and Philip S. Yu. 2024. Cosent: Consistent sentence embedding via similarity ranking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2800–2813.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *arXiv preprint arXiv:2004.12832*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

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

Annie Ma, Alex Yuxuan Peng, and Joseph Zhang. 2023. Fine-tuning bert for sentiment analysis, paraphrase detection and semantic textual similarity.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

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

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.

Li Im Tan, Wai San Phang, Kim On Chin, and Anthony Patricia. 2015. Rule-based sentiment analysis for financial news. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1601–1606.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Tianwen Wei, Jianwei Qi, and Shenghuan He. 2022. A flexible multi-task model for bert serving. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 785–796. Association for Computational Linguistics.

Gouki Yasui, Yoshimasa Tsuruoka, and Masaaki Nagata. 2019. Using semantic similarity as reward for reinforcement learning in sentence generation. In *Annual Meeting of the Association for Computational Linguistics*.

Haoxiang Zhang, Huaxiong Zhang, Xingyu Lu, and Qiang Gao. 2021. Attention-based overall enhance network for chinese semantic textual similarity measure.