minBERT and Downstream Tasks

Stanford CS224N Default Project

Xinpei Yu
xinpeiyu@stanford.edu

Abstract

This project focuses on advancing the minBERT model for sentiment analysis, paraphrase detection, and semantic textual similarity tasks. I explored multiple enhancement strategies, notably sharing weights between paraphrase detection and semantic textual similarity layers due to task similarities, optimizing loss functions—employing BCELoss for paraphrase detection while cosine similarity and customized Pearson correlation for semantic textual similarity. The effectiveness of these methods was rigorously compared, showing promising directions for model refinement. Future work will investigate multitask fine-tuning to leverage task synergies further.

1 Key Information to include

- Mentor: Rohan Taori (rtaori@cs.stanford.edu)
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

BERT (Bidirectional Encoder Representations from Transformers) has revolutionized the field of NLP, offering unprecedented capabilities in understanding and generating human-like text. This project leverages the minBERT, a more efficient variant of the original model, to tackle three crucial NLP tasks: sentiment analysis, paraphrase detection, and semantic textual similarity (STS).

Sentiment Analysis is foundational to text analysis, aiming to classify the polarity of given text as positive, negative, or neutral. This task is paramount in various applications, from gauging public sentiment towards products or political figures to understanding emotional undertones in large text corpora. The challenge lies in accurately capturing not just the explicit expressions of sentiment but also the subtleties and nuances inherent in human language.

Paraphrase Detection, on the other hand, focuses on identifying different expressions that convey the same meaning. This task is essential for systems to recognize and understand the diversity of human language expression, allowing for more efficient information retrieval, summarization, and question-answering systems. The primary difficulty in paraphrase detection is the ability to discern semantic equivalence amidst varied linguistic structures.

Semantic Textual Similarity (STS) extends beyond the binary outcomes of paraphrase detection to quantify the degree of semantic relatedness between texts. This nuanced understanding of textual similarity is crucial for tasks that require fine-grained semantic analysis, such as document clustering and information extraction. The challenge with STS is developing a model sensitive enough to capture varying degrees of similarity accurately.

To enhance the performance of the minBERT model on these tasks, I explored different extensions, including sharing weights between the layers dedicated to paraphrase detection and STS, and loss function optimization (employing BCELoss for paraphrase detection and experimenting with cosine similarity and customized Pearson correlation scores for semantic textual similarity). These strategies aim to refine the model's accuracy and effectiveness.

3 Related Work

The advent of BERT (Bidirectional Encoder Representations from Transformers), as introduced by Devlin et al. (2018) [1], marked a significant milestone in natural language processing. BERT's architecture leverages deep bidirectional representations, pretraining on unlabeled text by conditioning on both left and right context across all layers. This design enables BERT to be fine-tuned with minimal adjustments for a wide array of tasks, achieving state-of-the-art results on numerous benchmarks, including GLUE, MultiNLI, and SQuAD.

Building on BERT's foundation, subsequent research has explored various fine-tuning strategies to tailor these models to specific tasks. Notably, Sun et al. (2019) [2] discuss techniques for fine-tuning BERT for text classification, emphasizing the importance of domain-specific pretraining. This involves further pretraining on target-domain data using objectives such as the masked LM task, aligning the model more closely with the task-specific data distribution.

4 Approach

4.1 Model Architecture

My system is built upon a foundational minBERT base model, which is further fine-tuned for three tasks: sentiment analysis, paraphrase detection, and semantic textual similarity.

For Sentiment Analysis, the output from the base minBERT model is processed through a dropout layer to prevent overfitting, followed by a dense layer of size (`BERT_HIDDEN_SIZE`, `N_SENTIMENT_CLASSES`) to classify the sentiment of the input text into predefined categories.

In the case of paraphrase detection and semantic textual similarity, the model handles two input sentences, referred to as Sentence A and Sentence B. Each sentence is independently passed through the minBERT model, incorporating the pooler output, followed by a dropout layer and a dense layer of size (`BERT_HIDDEN_SIZE`, `BERT_HIDDEN_SIZE`). The outputs of these paths are then concatenated and passed through another dense layer of size (2 * `BERT_HIDDEN_SIZE`, 1) to make a final decision on whether the sentences are paraphrases (for paraphrase detection) or to determine the degree of semantic similarity (for semantic textual similarity).

Please refer to the graph in Figure 1 for a visual illustration of the model architecture.

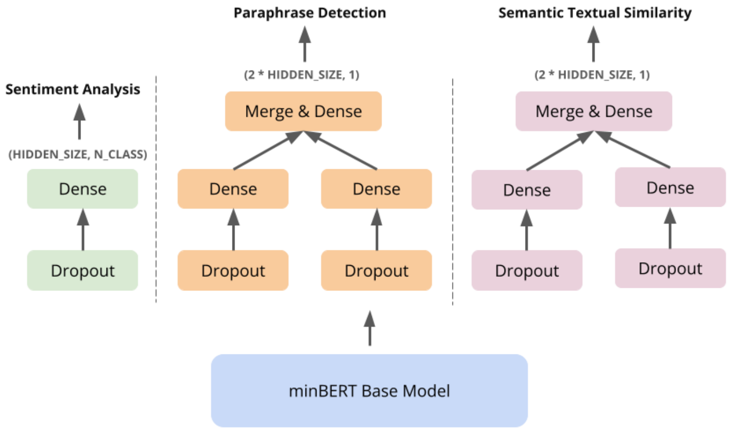

Figure 1: Baseline model architecture

4.2 Improvement 1: Sharing Weights

In an effort to enhance model efficiency and task synergy, I implemented a shared output layer for the paraphrase detection and semantic textual similarity (STS) tasks. This decision was informed by the inherent similarities between these two tasks, both of which involve assessing the relationship between pairs of sentences. By sharing the output layer, the model leverages commonalities in task structure to improve performance and reduce the total number of parameters. For a detailed visualization of this architectural improvement, please refer to the Figure 2 below.

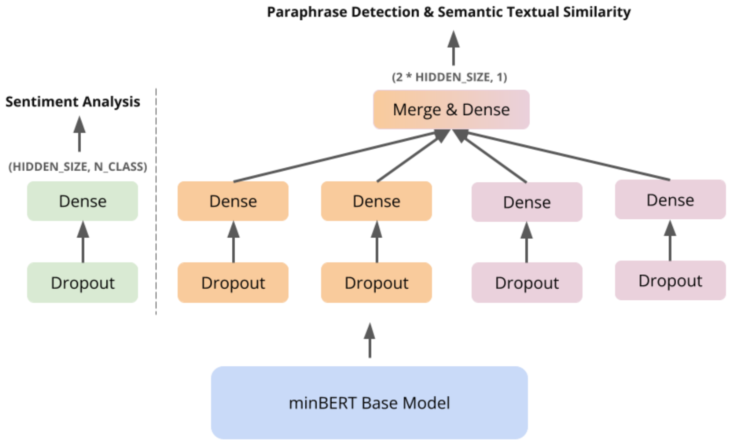

Figure 2: Sharing weights model architecture

4.3 Loss Functions

In the baseline model, the choice of loss functions is pivotal to aligning with the specific objectives of each task. For Sentiment Analysis, I employ Cross Entropy Loss, which is well-suited for classification tasks by measuring the dissimilarity between the true label distribution and the predictions. On the other hand, for paraphrase detection and semantic textual similarity (STS), I utilize L1 Loss, also known as Least Absolute Deviations. This loss function measures the absolute differences between the target values and the predictions, offering a simple yet effective approach for regression tasks where the goal is to predict continuous outputs.

The following table summarizes the loss functions employed for each task:

Task	**Loss Function**
Sentiment Analysis	Cross Entropy Loss
Paraphrase Detection	L1 Loss
Semantic Textual Similarity (STS)	L1 Loss

Table 1: Loss functions utilized for each task in the minBERT model.

4.4 Improvement 2: Loss Function Optimization

In pursuit of further refining model performance, I explored alternative loss functions for specific tasks. For paraphrase detection, Binary Cross Entropy Loss (BCELoss) was employed. BCELoss is particularly suited for binary classification tasks, as it measures the difference between two probabilities - the actual label and the predicted probability. This makes it a natural fit for paraphrase detection, where the task is to classify pairs of sentences as paraphrases or not.

For the semantic textual similarity (STS) task, I experimented with two different loss functions: Cosine Similarity and Pearson Correlation Score. Cosine Similarity measures the cosine of the angle between two vectors, which represent sentence embeddings. This metric is beneficial for STS as it effectively captures the semantic similarity between sentences. The Pearson Correlation Score, another measure of linear correlation between two sets of data, was also used to gauge the degree to which two sentences share semantic meaning. Both these metrics are expected to enhance the model's ability to discern nuanced semantic relationships, potentially leading to improved performance in the STS task.

The table below outlines the optimized loss functions applied to each task following these improvements:

Task	Optimized Loss Function
Sentiment Analysis (SST)	Cross Entropy Loss
Paraphrase Detection	BCELoss
Semantic Textual Similarity (STS)	Cosine Similarity / Pearson Correlation

Table 2: Optimized loss functions for each task in the enhanced minBERT model.

5 Experiments

5.1 Data

The model leverages 3 distinct datasets for training and evaluation across different tasks. For sentiment analysis, I utilize the Stanford Sentiment Treebank dataset, featuring movie review sentences labeled with varying degrees of sentiment. The Quora Question Pairs dataset serves the paraphrase detection task needs, containing labeled question pairs as paraphrases or not. Lastly, the semantic textual similarity task employs the SemEval STS Benchmark Dataset, which includes sentence pairs annotated with similarity scores.

The following table summarizes the datasets and their partitioning:

Task	Dataset	Training	Dev	Test
Sentiment Analysis	Stanford Sentiment Treebank	8,544	1,101	2,210
Paraphrase Detection	Quora Question Pairs	141,506	20,215	40,431
Semantic Textual Similarity	SemEval STS Benchmark	6,041	864	1,726

Table 3: Datasets used for training and evaluating the model across tasks.

5.2 Evaluation method

For sentiment analysis and paraphrase detection, I utilize *Accuracy* as the primary metric. Accuracy measures the proportion of correct predictions among the total number of cases examined, making it a straightforward and intuitive metric for classification tasks.

For the semantic textual similarity task, I employ the *Pearson Correlation Score* to evaluate performance. The Pearson Correlation Score quantifies the linear correlation between two variables, ranging from -1 to 1. A score of 1 implies a perfect positive linear relationship, -1 a perfect negative linear relationship, and 0 no linear relationship. This metric is particularly suited for the STS task as it effectively captures the degree to which predicted similarity scores align with human-annotated scores, reflecting the model's ability to understand nuanced semantic relationships.

5.3 Experimental details

The experiments were conducted in a fine-tuning mode, utilizing the pretrained minBERT model as the foundation. The key configurations for my experiments are as follows:

- **Mode:** Fine-tuning on the pretrained minBERT model.
- **Learning Rate:** Set at 1×10^{-5}, this learning rate was chosen to achieve a balance between fast convergence and avoiding overshooting the minima in the optimization landscape.
- **Epochs:** 10 epochs, a decision made to ensure sufficient training to reach optimal performance without overfitting.
- **Computational Resources:** The experiments utilized a NVIDIA GPU, which provided the necessary computational power to efficiently process the models and datasets.

5.4 Results

The following table summarizes the performance of model across different configurations and tasks, both on the development and test datasets.

	Sentiment Analysis		Paraphrase Detection		STS	
Model	Dev Acc	Test Acc	Dev Acc	Test Acc	Dev Corr	Test Corr
Baseline	0.456	-	0.732	-	0.309	-
Shared Weights	0.478	0.486	0.728	0.732	0.336	0.309
Cosine Model	0.449	0.453	0.725	0.727	0.274	0.293
BCELoss & Pearson	**0.457**	**0.455**	**0.780**	**0.788**	**0.331**	**0.294**

Table 4: Performance of different model configurations on development and test sets.

The final and activated submission to the TEST leaderboard is based on the results of the **BCELoss & Pearson** model. As of the night of March 16th, this submission, named `xp_new_loss`, achieved rank 100 with an overall score of 0.63.

6 Analysis

By inspecting model outputs and error patterns, I observed that the shared weights strategy between paraphrase detection and STS tasks tends to improve the model's ability to generalize across semantically related tasks. However, this approach also highlighted the challenge of balancing task-specific nuances within a shared framework.

The adoption of task-specific loss functions, particularly BCELoss for paraphrase detection and Pearson correlation for STS, demonstrated improvements. This suggests that aligning the loss function more closely with the task's nature and evaluation metrics can significantly enhance model performance.

Nevertheless, the system exhibits limitations in handling highly nuanced expressions of sentiment and complex paraphrases, indicating areas for future refinement. Furthermore, the variability in performance across different configurations underscores the importance of continuous experimentation and optimization in model development.

7 Conclusion

This project explored various strategies to enhance the minBERT model's performance on sentiment analysis, paraphrase detection, and semantic textual similarity tasks. Key improvements included the introduction of shared weights between related tasks and the optimization of loss functions to better align with each task's specific requirements.

My findings indicate that these enhancements contribute positively to model performance, with the BCELoss & Pearson model showing particularly promising results on the TEST leaderboard. Despite these advancements, the qualitative analysis reveals room for improvement, particularly in handling complex linguistic structures and nuanced expressions.

Future work could explore more sophisticated methods for task synergy, such as dynamic weight sharing, and investigate alternative loss functions that might offer further improvements in model 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] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *Chinese computational linguistics: 18th China national conference, CCL 2019, Kunming, China, October 18–20, 2019, proceedings 18*, pages 194–206. Springer, 2019.