Enhancing BERT through Multitask Fine-Tuning, Multiple Negatives Ranking and Cosine-Similarity

Stanford CS224N Default Project

Emily Broadhurst
Department of Computer Science
Stanford University
ebroad23@stanford.edu

Michael Maffezzoli
Department of Computer Science
Stanford University
maff2023@stanford.edu

Abstract

This study investigates the enhancement of BERT, a model in Natural Language Processing, specifically for text classification tasks where its full potential remains underexplored. Despite BERT's demonstrated efficacy across a spectrum of NLP tasks through its bidirectional training of transformers, its applicability to text classification, paraphrase detection and sentence similarity detection has been comparatively limited. The research presented here aims to extend BERT's multitask utility by incorporating multitask fine-tuning, cosine similarity fine-tuning, and multiple negatives ranking loss learning, to enhance performance across these tasks. We utilized three datasets to perform this study: SemEval, Stanford Sentiment Treebank, and Quora. Our findings prove these methods result in a significant improvement in multitask performance over the conventional BERT model. This advancement facilitates more accurate sentiment analysis, paraphrase detection, and semantic textual similarity tasks. Future work could explore the integration of advanced pre-training methods to further improve multitask performance.

1 Key Information to include

- Mentor: David Lim
- Contributions:
  1. Emily: Implemented cosine similarity and multiple negatives ranking loss learning.
  2. Michael: Implemented optimizer and multitask classifier. Ran GCP training experiments.

2 Introduction

In this project, we implemented the basics of a BERT model, and performed three tasks: sentiment analysis (SST), paraphrase detection (Quora), and semantic textual similarity (SemEval STS). The introduction of machine learning models, particularly Bidirectional Encoder Representations from Transformers (BERT), has significantly advanced the field of Natural Language Processing (NLP). Despite BERT's achievements, its application in generating robust, generalizable sentence embeddings for text classification tasks such as sentiment analysis, paraphrase detection, and semantic textual similarity requires further exploration. This project aims to enhance BERT's performance on these tasks by employing a multifaceted approach to fine-tuning and loss strategies.

Understanding language is a complex task due to its contextual variability and semantic depth. Current, models find it difficult to capture the subtleties necessary for accurate and generalizable performance across contexts. To perform well, a model has to have seen the word before, which presents challenges around idioms, slang, and more obscure vocabulary. Extending BERT allows for a more nuanced understanding of the language and thus more applications where tone, context, and meaning are significant—our three goal tasks. This work has wide-ranging potential applications such as in finance, and healthcare, amongst other industries.

This study tested whether a combination of multitask learning, cosine-similarity fine-tuning, and multiple negatives ranking loss boosts BERT multitask performance. These methods are designed to refine BERT's understanding of sentence relationships, improving its performance in sentiment analysis, paraphrase detection, and semantic textual similarity. Multitask learning is employed to leverage the relatedness of the selected tasks, aiming to foster a more generalized representation capability within BERT. Cosine-similarity fine-tuning is intended to enhance the model's ability to assess semantic closeness, crucial for tasks involving paraphrase detection and textual similarity. The use of multiple negatives ranking loss aims to improve the model's ability to differentiate between similar but distinct sentences. In summary, we hope to enhance BERT to create more accurate NLP tools for sentiment analysis, paraphrase detection and sentence similarity.

3 Related Work

Devlin et al. pioneered the development of the BERT model, a transformative approach to pre-training bidirectional representations from textual data by concurrently considering context from both left and right directions across all layers (Devlin et al., 2019). This innovative methodology significantly propelled advancements in the field of natural language understanding, demonstrating marked improvements in performance across a variety of NLP tasks, including but not limited to, question answering, named entity recognition, and sentiment analysis. Building upon this foundational work, Liu et al. further refined the BERT model through the introduction of RoBERTa, which involved extensive additional training and modifications to the original masking strategy (Liu et al., 2019). This enhancement underscored the potential of optimizing pre-training techniques, setting new benchmarks in model efficacy, and opening avenues for further research in deep learning-based NLP models.

Thus, we will use these methods as a foundation to extend BERT for multitask learning on sentiment analysis, paraphrase detection, and semantic textual similarity.

4 Approach

Initially, we implemented BERT according to the specifications of the original BERT paper (Devlin et al., 2019). Subsequently, we enhanced our baseline model by employing multitask fine-tuning, integrating cosine similarity, then testing the performance of different loss functions: binary cross entropy, multiple negatives ranking, and mean squared error.

4.1 BERT Baseline

In this project, we adhere to the specifications outlined in the original paper for implementing BERT (Devlin et al., 2019). Our initial step involves converting sentence inputs into tokens using the WordPiece tokenizer. This method breaks down each word into one of 30,000 ids. In this preprocessing phase, words that are not recognized are marked with the [UNK] token, and to standardize the length of all sentences, we append the [PAD] token as necessary. Additionally, BERT introduces the [SEP] token to distinctly separate sentences.

Following tokenization, BERT employs an embedding layer to convert each token into a vector representation. Subsequently, the architecture utilizes several encoder transformer layers that feature attention mechanisms, mathematically represented as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

The multi-head self-attention is calculated by finding the scaled-dot product of queries and keys, normalized by the square root of the dimensionality of the keys (d_k), and applying a softmax function to ascertain the weights. This mechanism is formalized in the following equations:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2}$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

Each transformer layer is comprised of two linear transformations followed by a ReLU activation function, as indicated by:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{4}$$

To enhance model generalization, dropout is applied to each sub-layer prior to its addition to the sub-layer input and subsequent normalization. The training for BERT includes a proportion of the tokens being masked for the model to predict, utilizing pre-trained model weights and output embeddings for applications such as sentiment analysis. For optimization, we employed an Adam Optimizer with Decoupled Weight Decay Regularization, which is efficient for stochastic optimization and requires only first-order gradients. Then, we enhanced BERT by leveraging cosine-similarity fine-tuning, multiple negatives ranking loss learning, and multitask fine-tuning.

4.2 Multitask Fine-tuning

We implemented a multitask fine-tuning approach for the BERT model, diverging from the conventional method of task-specific fine-tuning. Rather than updating BERT's parameters separately for each task, we employed a multi-task learning strategy that simultaneously updates the model based on multiple tasks. Within each epoch, we loop through all the batches of each dataset simultaneously, combining the losses at each iteration and updating based on that combined loss. Building on that we utilized work by Lin et al. which described a technique for combining the losses, taking the weighted average of the task losses and assigning the weights randomly for each iteration. The loss weights are sampled from a normal distribution, followed by the application of normalization to these weights. The model's parameters are then updated by minimizing this combined loss (Lin et al., 2022). This approach is designed to reduce the likelihood of converging to local minima, thereby potentially improving the model's generalization capabilities across tasks. The total loss is defined as the weighted sum of the individual task losses:

$$\begin{aligned} L_{\text{total}} = W[0] \times L_{\text{sent}} + W[1] \times L_{\text{para}} + W[2] \times L_{\text{sts}} \\ \text{where } W = \text{list of three random numbers} \end{aligned} \tag{5}$$

4.3 Muliple Negatives Ranking Loss Learning

Next, we tested enhancing BERT with Multiple Negatives Ranking Loss (Henderson et al., 2017). Specifically, we examined sets of K sentence pairs, denoted as $[(a_1, b_1), \ldots, (a_n, b_n)]$. Here, a_i and b_i represent pairs of similar sentences, and (a_i, b_j) where $i \neq j$ are deemed dissimilar. The goal of the Multiple Negatives Ranking Loss is to widen the distance between dissimilar sentence pairs (a_i, b_j) for $i \neq j$, while narrowing the distance between similar pairs (a_i, b_i). The loss function for a single batch is given by:

$$J(x, y, \theta) = -\frac{1}{K}\sum_{i=1}^{K} \log P_{\text{approx}}(y_i|x_i) = -\frac{1}{K}\sum_{i=1}^{K}\left[S(x_i, y_i) - \log\sum_{j=1}^{K} e^{S(x_i, y_j)}\right], \tag{6}$$

In this equation, θ signifies the embedding and neural network parameters that contribute to the scoring function S.

4.4 Cosine Similarity

Our third candidate enhancement of BERT involved fine-tuning based on cosine similarity. This approach computes the similarity between embeddings using the cosine of the angle between them. During the fine-tuning process, we used MSE loss between the cosine similarities and the labels scaled to the same range $[-1, 1]$. According to this metric, sentences with high similarity yield a cosine similarity score close to 1, whereas dissimilar sentences result in a score closer to -1 (Reimers and Gurevych, 2019).

5 Experiments

To enhance BERT, we implemented Multitask Fine-Tuning, Multiple Negatives Ranking, and Cosine Similarity optimization in a staggered and systematic manner. We integrated these enhancements sequentially, assessing their impact on model performance against a baseline. Approaches that improved performance were retained for further experimentation. This section describes the experiments conducted, detailing how we applied each method and its effect on performance. Through this process, we identified the most effective strategies for improving BERT, aiming for superior multitask performance.

5.1 Data

In our study, we utilized three datasets: the Stanford Sentiment Treebank (SST), SemEval STS Benchmark Dataset and the Quora Dataset. The SST comprises 11,855 sentences extracted from movie reviews, with each sentence labeled as negative, somewhat negative, neutral, somewhat positive, or positive, facilitating sentiment analysis (Rec). The Quora Dataset consists of 400,000 question pairs, each labeled to indicate whether the pairs are paraphrases of each other, serving our paraphrase identification task (Quo). Additionally, we employed the SemEval STS Benchmark Dataset, which contains 8,628 sentence pairs annotated with similarity scores ranging from 0 to 5, to further support our evaluation of semantic similarity measures (Rec).

- SST (Stanford Sentiment Treebank) dataset for sentiment analysis

  Input: ID of the sentence and the sentence itself.

  Output: The sentiment of the sentence, rated on a five-point sentiment scale (2 is neutral).
- Quora dataset for paraphrase detection

  Input: ID of the sentence pair and the sentence pair itself.

  Output: Prediction of whether the sentences in the pair are paraphrases of each other (binary).
- SemEval dataset for semantic textual similarity (STS)

  Input: ID of the sentence pair and the sentence pair itself.

  Output: A similarity score indicating the degree of semantic similarity between the two sentences from 0 to 5.

5.2 Evaluation method

The evaluation metrics for paraphrase detection and sentiment classification tasks are based on accuracy, which is calculated as $\frac{\text{number of correct predictions}}{\text{total number of predictions}}$. Furthermore, the task of semantic textual similarity is assessed through the Pearson Correlation Coefficient, ranging from -1 to 1, to quantify the linear correlation between the actual and predicted labels. To provide a comprehensive evaluation of multitask performance, we normalize the scores to $[0, 1]$ and then compute the average of these three metrics.

5.3 Experimental details

We conducted six experiments outlined below.

1. **Baseline**: Pretrained from `bert-base-uncased`. Fine-tuned on sentiment classification with the SST database, using cross-entropy loss. The training was conducted over 10 epochs with a learning rate of 1×10^{-5}, a batch size of 8, and a hidden dropout probability of 0.3 (the lr, batch size, and dropout were the same for all experiments).
2. **Multi-task Fine-tune**: Pretrained from `bert-base-uncased`. Fine-tuned on all three datasets (SST, Quora, STS), employing cross-entropy, binary cross-entropy, and mean squared error losses, respectively, for each dataset. The sentiment embeddings were passed through a dropout and one linear layer. The paraphrase and similarity embeddings were concatenated then passed through a dropout and one linear layer. This setup utilized the previously mentioned random loss weighting technique. Each dataset was standardized to

12080 examples (2x the smallest dataset), either by duplicating entries in smaller datasets or sampling from larger ones. The model was trained for 6 epochs with a learning rate of 1×10^{-5}.

3. **MNR Loss for Paraphrase Detection and STS**: Pretrained from `bert-base-uncased`. Fine-tuned on the SST, Quora, and STS datasets, using cross-entropy for SST and multiple negative ranking (MNR) loss for the Quora and STS datasets. This configuration also incorporated the aforementioned random loss weighting method and also used 12080 examples from each dataset. Training was carried out over 6 epochs with a learning rate of 1×10^{-5}.
4. **Cosine Similarity for STS**: With this experiment we passed the cosine similarity of the pooled bert embeddings into MSE loss, scaling the labels to $[-1, 1]$. This replaced the concatenation followed by a dropout and linear layer. The other losses and parameters were the same as experiment 3.
5. **BCE for Para and Cosine Similarity for STS**: Pretrained from `bert-base-uncased`. Fine-tuned on all three datasets (SST, Quora, STS), employing cross-entropy loss, binary cross-entropy loss, and mean squared error loss of cosine similarities, respectively. Run with the same parameters as the previous 3 experiments.
6. **More Finetuning**: With the same losses from experiment 5, we trained on 50000 examples per dataset for 10 epochs with the same lr and batch size as all other experiments.

5.4 Results

Table 1: Development Set Performance

Experiment	SST Dev Accuracy	Para Dev Accuracy	STS Dev Correlation	Overall
1	0.509	0.509	0.031	0.511
2	0.471	0.721	0.344	0.621
3	0.505	0.675	0.272	0.605
4	0.490	0.648	0.433	0.618
5	0.505	0.730	0.471	0.657
6	0.488	0.757	0.519	0.668
6 (test scores)	0.515	0.757	0.509	0.676

As expected, multitask finetuning (exp. 2) greatly improved the models performance on the two tasks it was not finetuned on in experiment 1. We were surprised to see that MNR loss (exp. 3) decreased the models paraphrase accuracy and similarity correctness as we considered MNR a more robust loss for sentence comparison tasks. Cosine similarity is a more intuitive way to handle the embeddings for the similarity task than concatenation, so even without the dropout and linear layer, this method produced better results for that task. As expected, once we combined the two losses that performed best for the paraphrase and similarity tasks, we got our best results (exp. 5). Finally, we were surprised that greatly increasing the number of training examples and epochs did not have a significant effect on the overall performance on the tasks.

6 Analysis

To better understand our system's performance, we analyzed accuracy changes across six experiments. This section provides a qualitative evaluation aimed at identifying insights to enhance our BERT model's effectiveness. We focused on understanding the reasons behind accuracy fluctuations to guide future improvements.

1. **Baseline**:
Interestingly, when fine-tuned solely on SST data, the BERT model shows around 50 percent accuracy for both similarity and paraphrase tasks. This may indicate the model's strong classification ability, although it struggles more with similarity tasks, which are not purely classification-based. Upon inspecting SST predictions, we noted the model's difficulty in distinguishing between sentences that are somewhat positive and those that are positive, a task that appears subjective and challenging. For example, the sentence "It's a lovely film

with lovely performances by Buy and Accorsi" received an incorrect prediction (overshot by 1), likely influenced by the repeated use of "lovely." Similarly, the model misclassified "And if you're not nearly moved to tears by a couple of scenes, you've got ice water in your veins" as positive, possibly confused by idiomatic language. These errors highlight the challenges in differentiating between subtle sentiment levels. However, the model did correctly identify some positive sentiments, like "A warm, funny, engaging film" suggesting a possible bias towards predicting positive sentiments over somewhat positive ones.

2. **Multi-Task Fine-Tune**:
   In our study, we adopted a multi-task fine-tuning approach, wherein each training batch consisted of alternating tasks. This methodology led to significant enhancements in paraphrase detection accuracy and similarity scoring, albeit with a slight decline in sentiment analysis accuracy. Such outcomes suggest that the model is improving in paraphrase and similarity tasks while attempting to retain its performance on sentiment analysis.

   Looking into the prediction data, the model accurately recognized the paraphrase relationship between "How do I learn any language fast?" and "How can one learn a new language quickly?", correctly identifying them as similar. Conversely, it misclassified the pair "What are the most inspiring/motivational books about life?" and "What is the most powerful tip you've gained reading a self-help book?" This error may indicate that the multi-task fine-tuning process overly depends on lexical similarities, failing to grasp syntactical distinctions. This suggests a deficiency in the model's ability to discern nuanced differences, potentially due to an overemphasis on pattern recognition in its training regimen.

3. **MNR Loss for Paraphrase Detection and STS**:
   Contrary to our expectations, employing a more complex loss function, specifically multiple negative ranking (MNR), did not enhance our model's performance; rather, it slightly deteriorated. This suggests that the increased complexity of the learning process introduced by MNR might not be necessary, as the paraphrase detection task was adequately addressed with binary cross-entropy loss. Moreover, the similarity task appeared to align more naturally with a regression framework (utilizing mean squared error (MSE) loss) rather than transforming continuous labels into discrete classifications.

   The observed trend from the previous experiment, where the model disproportionately emphasized vocabulary over semantics, was also evident in the similarity task. It appears that MNR excessively focuses on keyword detection instead of capturing semantic relationships. For example, the model incorrectly identified the sentences "What can you get as a customer of Star Alliance?" and "What are some ways to register with Star Alliance?" as similar, despite their semantic dissimilarity. Conversely, it correctly recognized the similarity between "Why is the sociology of education important to a teacher" and "How is sociology of education helpful to a teacher?", which suggests a reliance on word matching rather than a deep semantic understanding. This demonstrates a significant model limitation, highlighting the need for approaches that better capture the nuances of semantic relationships beyond mere lexical similarities.

4. **Cosine Similarity for STS**:
   In our study, we explored the use of mean squared error (MSE) loss between the cosine similarity of sentence embeddings and their labels to evaluate sentence similarity, which significantly enhanced our Semantic Textual Similarity (STS) scores. This improvement is attributed to the model's direct training on the cosine similarity of embeddings, as opposed to a concatenation-based approach. However, the model's failure to classify semantically similar sentences, such as "Two black dogs are playing on the grass" and "Two black dogs are playing in a grassy plain," raises concerns about potential biases and limitations. These include a focus on specific vocabulary, constraints of contextual embeddings, insufficiently diverse training data, architectural limitations, and possibly inadequate evaluation metrics and loss functions. Such issues suggest the model's struggle to move beyond mere lexical similarities to capture deeper semantic meanings, a challenge that might stem from a training set lacking diversity or an architecture ill-suited for nuanced semantic comprehension. To overcome these hurdles, enhancing the training dataset with more varied examples, investigating advanced embedding techniques or architectures, and refining evaluation metrics to more accurately reflect semantic understanding are crucial steps

5. **BCE for Para and Cosine Similarity for STS**:
   This experiment merged the two most effective loss functions for paraphrase and similarity tasks, leading to anticipated enhancements across all three tasks. The improvement suggests that optimizing the loss function for one task can positively influence the model's performance on other tasks as well. Given that this approach was based on consolidating findings from previous experiments, no additional qualitative evaluation was necessary for this phase, as it did not yield novel insights beyond what was already established.
6. **More Fine-tuning**:
   Subsequent iterations yielded marginally improved accuracy, suggesting that earlier model runs may have been prematurely terminated.

7 Conclusion

In this project, we explored the effects of extending a BERT pre-trained model to generalized multitask performance. We first implemented multitask finetuning and then experimented with several loss functions for each task: binary cross-entropy, multiple negative ranking loss, mean squared error loss of concatenated embeddings, and mean squared error loss of cosine similarity.

Our experiments showed that multitask fine-tuning greatly increased the model's performance on the paraphrase and similarity tasks compared to just fine-tuned on sentiment data, while having a minimal negative effect on the sentiment task accuracy. Our experiments illustrate that binary cross entropy loss outperformed MNR for the paraphrase detection task, and MSE loss of the cosine similarity of the direct BERT embeddings outperformed MNR for the sentence similarity task. Increasing the number of examples by almost 5x and almost doubling the number of epochs did not have a large impact on the overall performance of the model.

Future work could enumerate all combinations of different loss functions for all 3 tasks; three different losses for each task and thus 27 experiments. There is also an opportunity to investigate how to combine the losses of the three tasks to achieve the best learning; we used random weighting but there are many other approaches that could yield better results. Furthermore, pre-training with target-domain data would additionally enhance BERT performance. However, due to the limitations on our time this quarter we were not able to implement these methods.

References

Deep models for semantic compositionality over a sentiment treebank.

First quora dataset release question pairs.

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

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply.

Baijiong Lin, Feiyang YE, and Yu Zhang. 2022. A closer look at loss weighting in multi-task learning.

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

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.