Minh-BERT

Stanford CS224N Default Project

Taran Kota
Department of Computer Science
Stanford University
tkota@stanford.edu

Abstract

In this research, we focus on BERT due to its groundbreaking ability to understand bidirectional context as a transformer model. The two main goals of our project are to develop a streamlined version of BERT (minBERT) for sentiment analysis, and to evaluate minBERT's performance in sentiment analysis and other related tasks like paraphrase detection and semantic textual similarity (STS). We complete a baseline model for minBERT in the first goal, and enhance this model with several extensions such as For the second goal, we enhanced minBERT with several advanced techniques such as cosine similarity, gradient surgury, additional datasets, and implementing a multi-attention network.

1 Key Information to include

I originally took CS224N in the Winter of 2024, but took an incomplete in the class. This project was completed last quarter, and it is only the final report and poster presentation that I cam completing this quarter. My leaderboard submission is currently in the Winter 23-24 gradescope.

2 Introduction

From product reviews to social media comments, our society increasingly interacts through text-based communications online. These interactions occur on a massive scale, and the text processing techniques used to analyze this data are beginning to significantly influence how companies, managers, and designers develop products for consumers. In recent years, advancements in the natural language processing (NLP) field—particularly transformer-based models that generate dynamic, context-aware representations of large datasets—have enabled language models to perform sophisticated analytical tasks. This study focuses on three downstream tasks: (1) sentiment analysis, which classifies text into five sentiment classes ranging from negative to positive, (2) paraphrase detection, which determines whether two distinct text segments convey the same meaning, and (3) similarity analysis, which assesses whether two text segments are more similar to each other than not. The successful execution of these tasks has significant economic implications, particularly for review aggregation software.

To achieve this, we utilized the canonical Bidirectional Encoder Representations from Transformers model, known as "BERT". According to Devlin et al. (2018), the BERT model can be fine-tuned for various tasks without requiring substantial task-specific architectural modifications. While our implementation of the baseline BERT model accurately predicted sentiment over half the time, its performance on the other two tasks did not exceed this threshold. This motivated us to extend the baseline BERT model's functionality to improve its performance across all three tasks.

The remainder of the paper talks about related works to this project (Section 3), the implementation details of the baseline BERT model as well as the additional extensions (Section 4), and the experimental results and analysis (Sections 5 and 6).

3 Related Work

3.1 Cosine Similarity

Cosine similarity loss has emerged as a critical technique in the realm of natural language processing, particularly for tasks requiring the evaluation of semantic similarity between textual representations. This loss function measures the cosine of the angle between two non-zero vectors, effectively capturing the degree of similarity in their orientation, which translates well into understanding semantic similarity. Notably, it has been leveraged in models such as Sentence-BERT (Reimers and Gurevych, 2019), which modifies the original BERT architecture to better handle sentence embeddings by directly optimizing for cosine similarity during training. By doing so, it enables more accurate performance in downstream tasks like paraphrase identification and semantic textual similarity. The introduction of a dynamic margin in the cosine similarity loss further enhances its effectiveness, allowing the model to adaptively penalize varying degrees of dissimilarity, thereby refining its learning process. This approach has proven beneficial in multi-task learning scenarios, where maintaining a balance between multiple, potentially conflicting tasks is crucial. Integrating cosine similarity loss into minBERT aims to harness these advantages, enhancing its ability to perform nuanced sentiment analysis and other related tasks.

3.2 Gradient Surgery

Gradient surgery, a novel technique in the field of multi-task learning, has gained attention for its ability to manage and optimize conflicting gradients during the training process. This approach addresses a common challenge in multi-task learning: the presence of competing objectives, where gradients from different tasks may interfere with each other, leading to suboptimal performance. Gradient surgery, as introduced by Yu et al. (2020) in their work on gradient surgery for multi-task learning, involves modifying the gradient vectors to ensure that they are mutually compatible. By projecting the gradients onto a shared subspace or selectively adjusting their magnitudes, this method minimizes negative interference and promotes synergistic learning. This technique has shown significant improvements in scenarios where tasks have divergent or conflicting goals, such as in natural language processing and computer vision applications. For minBERT, implementing gradient surgery aims to optimize the model's performance across various downstream tasks, ensuring that enhancements in one task do not detrimentally impact others. By effectively managing the gradients, gradient surgery helps maintain a balanced learning trajectory, facilitating more robust and generalized model performance.

3.3 Aspect Based Sentiment Analysis

Aspect-Based Sentiment Analysis (ABSA) is a subfield of sentiment analysis that focuses on identifying the sentiment expressed towards specific aspects within a text. Previously, BERT has been combined with the Interactive Attention Networks (IAN) model by Ma et al. (2017) to improve the accuracy of aspect-based sentiment analysis. However, this approach heavily relied on manually tagging user reviews according to predefined aspects, which is inherently laborious, time-consuming, and biased.

More recently, Qiang et al. (2020) introduced a new Multiple-Attention Network (MAN) approach that achieves more effective ABSA without the need for aspect tags. This technique utilizes a Position-aware Attention submodule to capture the relevance of contextual words, enhancing the model's ability to understand and analyze sentiments related to specific aspects within a text.

4 Approach

4.1 Implementing Baseline minBERT

For the first step, we follow the steps in the default final project handout in this order:

1. The core of BERT's functionality lies in its multi-head attention mechanism and transformer layers. The multi-head attention, implemented in the BertSelfAttention class, allows the model to attend to different parts of the input sequence simultaneously, capturing diverse

aspects of contextual relationships. Each attention head processes the input independently, and their outputs are concatenated and transformed, providing a rich, multi-dimensional representation of the text. This mechanism is crucial for understanding and interpreting the nuances in language.

The transformer layers, implemented in the BertLayer class, consist of both the multi-head attention mechanism and feed-forward neural networks. These layers are designed to process and transform the contextual embeddings generated by the attention heads. The self-attention mechanism within the transformer layer enables the model to capture long-range dependencies in the text, while the feed-forward networks further refine these representations, enhancing the model's ability to understand complex linguistic structures. By leveraging these sophisticated components, BERT achieves a high level of performance in natural language processing tasks.

4.2 Cosine Similarity

To enhance the performance of our minBERT model, we implemented cosine similarity as part of our training objective. The cosine similarity metric measures the cosine of the angle between two vectors, providing a measure of their similarity irrespective of their magnitude. This is particularly useful for tasks requiring fine-grained semantic similarity, such as Semantic Textual Similarity (STS).

The cosine similarity between two vectors $\mathbf{u}$ and $\mathbf{v}$ is defined as:

$$\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$$

Here, $\mathbf{u} \cdot \mathbf{v}$ denotes the dot product of the vectors, and $\|\mathbf{u}\|$ and $\|\mathbf{v}\|$ represent their magnitudes (or Euclidean norms).

To integrate cosine similarity into our training objective, we define a loss function that encourages the model to produce embeddings that are close to each other for similar sentences. Specifically, we use a cosine-similarity embedding loss with a dynamic margin. The loss function L is formulated as follows:

$$L = 1 - \cos(\theta) = 1 - \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$$

This loss function penalizes embeddings that are not similar by subtracting the cosine similarity from 1, ensuring that the loss decreases as the similarity between the embeddings increases.

In practice, during training, we compute the embeddings for sentence pairs and apply the cosine similarity loss. This approach helps our minBERT model to learn more semantically meaningful embeddings, improving its performance on downstream tasks such as sentiment analysis and STS.

4.3 Gradient Surgury

To enhance the performance of our minBERT model on multiple tasks, we implemented gradient surgery as part of our multi-task learning strategy. Gradient surgery is a technique designed to address conflicts between gradients of different tasks during training, thereby improving the overall performance of the model.

Gradient surgery works by projecting the gradient of one task onto the normal plane of another task's gradient. This projection ensures that the gradients of conflicting tasks do not interfere with each other, allowing for more effective learning across multiple tasks.

Given two tasks with gradients $\mathbf{g}_1$ and $\mathbf{g}_2$, the gradient surgery process can be described as follows:

1. Compute the dot product of the gradients:
$$d = \mathbf{g}_1 \cdot \mathbf{g}_2$$
2. If $d < 0$, indicating conflicting gradients, project $\mathbf{g}_1$ onto the normal plane of $\mathbf{g}_2$:
$$\mathbf{g}_1^{proj} = \mathbf{g}_1 - \frac{d}{\|\mathbf{g}_2\|^2}\mathbf{g}_2$$

3. Use the modified gradient $\mathbf{g}_1^{proj}$ for updating the model parameters.

In our implementation, we apply gradient surgery during the backpropagation step of our training process. For each pair of tasks, we check for gradient conflicts and adjust the gradients accordingly. This approach helps our minBERT model to better handle multi-task learning, especially when tasks may conflict with each other.

By incorporating gradient surgery, we aim to improve the robustness and accuracy of our model across various downstream tasks, including sentiment analysis, paraphrase detection, and semantic textual similarity (STS).

4.4 Multiple-Attention Networks

Due to the complexity of our tasks, we decided to add a framework of linear layers as described in Qiang et al. (2020). This framework is incorporated after obtaining the output hidden layers from the BERT model and is used to produce more nuanced token embeddings for each task. This process includes three main steps: (1) self-attention, (2) position-aware attention, and (3) regularization and concatenation.

- We calculate self-attention using the output hidden layers from BERT and their transformations, similar to how self-attention is computed within the BERT model itself. This term is calculated for four predefined aspects. For each aspect, weighted outputs are calculated using a linear layer that updates the weights. The self-attention mechanism can be represented as:

$$\mathbf{A}_{self} = softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$$

  where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the query, key, and value matrices derived from the BERT hidden layers.

- The outputs from the self-attention calculation are then multiplied by a position-aware weighting factor for each aspect. This factor incorporates an average of the word, position, and token embeddings to capture the global context of the sentence, as shown in Qiang et al. (2020) and R. He and Dahlmeier (2017). The position-aware attention is calculated as follows:

$$\mathbf{A}_{pos} = softmax\left(\frac{\mathbf{Q}_{pos}\mathbf{K}_{pos}^T}{\sqrt{d_k}}\right)\mathbf{V}_{pos}$$

  Here, $\mathbf{Q}_{pos}$, $\mathbf{K}_{pos}$, and $\mathbf{V}_{pos}$ are the query, key, and value matrices that incorporate position-aware embeddings.

- Regularization terms are calculated to encourage orthogonality between attention weights on different aspects. These terms are incorporated into the task loss function. The results from each aspect's attention mechanism are concatenated to produce a comprehensive analysis of the sentence, which is subsequently used to generate predictions. The regularization term can be represented as:

$$\mathcal{L}_{reg} = \sum_{i,j,i\neq j} (\mathbf{W}_i \cdot \mathbf{W}_j)^2$$

  where $\mathbf{W}_i$ and $\mathbf{W}_j$ are the attention weight matrices for different aspects. This term ensures that the attention weights for different aspects remain orthogonal.

  By incorporating these steps, our MAN framework aims to enhance the performance of the minBERT model on various tasks, leveraging nuanced token embeddings and contextual information for better predictions.

5 Experiments

5.1 Data

We are using the Stanford Sentiment Treebank (SST) dataset for sentiment analysis and the Quora dataset for paraphrase detection. The SST dataset consists of 11,855 single sentences extracted from movie reviews. The dataset was parsed with the Stanford parser and includes a total of 215, 154 unique phrases, annotated by 3 human judges. Each phrase has a label of negative, somewhat negative, neutral, somewhat positive, or positive. The Quora dataset consists of 400, 000 question pairs with labels indicating whether particular instances are paraphrases of one another. We use a subset of the dataset with 202, 152 question pairs.

For STS, we use a combination of the SemEval and Sentences Involving Compositional Knowledge (SICK) datasets. The SemEval dataset contains 8631 sentence pairs that are rated on a continuous scale from 0-5 on how similar they are. Here 5 means both sentences have an equivalent meaning, whereas 0 means they are unrelated. The SICK dataset includes a large number of sentence pairs that are rich in the lexical, syntactic and semantic phenomena, and is rated in the same manner. The SICK dataset originally comes in a split of 4439 pairs in the train set, 495 in the trial set used for development and 4906 in the test set. We simply put all of the train and dev examples in the overall STS train set.

5.2 Evaluation method

For each of the three tasks, we consider the metrics provided on the dev and test leaderboards by the CS224N teaching team. Each time we add an extension, we consider the performance against the baseline milestone results as well as previous extensions to see whether we observe an improvement.

5.3 Experimental details

We use the hyperparameters suggested by the default handout and starter code for all experiments, which includes:

- 10 epochs
- Batch size of 9
- Dropout parameter of 0.3
- Pretraining learning rate of 1×10^{-3}
- Finetuning learning rate of 1×10^{-5}

Additionally, for MAN, we used an aspect size of four and $\lambda = 0.5$

5.4 Results

Below are the table of results that we attained with our extended minBERT model:

Metric	SST Accuracy	Paraphrase Accuracy	STS Correlation
Baseline	0.412	0.7900	0.568
Test Results	0.485 (+0.485)	0.818 (+0.818)	0.624 (+1.624)
Dev Results	0.508 (+0.000)	0.816 (+0.000)	0.619 (+0.000)

Table 1: Performance metrics for sentiment analysis (SST), paraphrase detection, and semantic textual similarity (STS) on test and development sets.

6 Analysis

Overall, I found that adding more data and implementing cosine similarity and gradient surgery resulted in the most significant improvement on the minBERT baseline model, since performance in

all 3 categories increased. The implementation of MAN resulted in improvements in some categories but performance decreases in others. This is where I think the model became over optimized for the paraphrase detection task, most likely because of the sheer amount of data in the Quora dataset. However, based on the results, we can see that we have implemented a functioning multiple-attention network.

7 Conclusion

In this study, we examined various extensions to the BERT model for three downstream tasks: sentiment analysis, paraphrase detection, and similarity analysis. Our extensions aimed to improve generalizability across all tasks while enhancing task-specific performance.

By changing loss functions and adding additional training data, we were able to improve the performance of the model. The implementation of Multi-Attention Networks (MAN) further enhanced the prediction of sentiment and similarity, resulting in a more sophisticated BERT model.

We learned that diagnosing poor accuracies can be challenging, especially after implementing multiple novel extensions.

Future work should focus on optimizing the model's computational efficiency and scalability, and increasing transparency in the model's decision-making processes through visualization of attention weights. Experimenting with MAN-related hyperparameters and implementation tweaks could also enhance specific task performance without adversely affecting others.

8 References

- Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. **sem 2013 shared task: Semantic textual similarity.*
- Yige Xu, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019.*
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv.*
- Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. *arXiv.*
- Yao Qiang, Xin Li, and Dongxiao Zhu. 2020. Toward tag-free aspect based sentiment analysis: A multiple attention network approach. *arXiv.*
- H. T. Ng, R. He, W. S. Lee, and D. Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).*
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems.*
- Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *arXiv.*