Enhancing BERT with Self-Supervised Attention

Stanford CS224N Default Project

Josh Francis Department of Computer Science Stanford University josfran@stanford.edu

Abstract

BERT's self-attention mechanism struggles with capturing long-range dependencies which has prompted modifications like sparse and global attention. In this paper we explore self-supervised attention (SSA) as an alternative approach, inspired by "Improving BERT with Self-Supervised Attention." We recreate the SSA mechanism, use BertViz for visualization, and evaluate our model on sentiment analysis, paraphrase detection, semantic textual similarity tasks, and the SQuAD benchmark. Our findings reveal that SSA improves performance when fine-tuning on a single task, but not in multitask learning scenarios, suggesting that alternative attention mechanisms may be more suitable for enhancing performance in multitask settings.

1 Introduction

BERT has achieved state-of-the-art results in various natural language processing tasks, addressing a wide range of challenges. However, capturing long-range dependencies in text, which are essential for tasks such as sentiment analysis, paraphrase detection, and semantic textual similarity, still remains a challenge for natural language processing. To tackle this issue, researchers have proposed several modifications to BERT's self-attention mechanism such as sparse attention and global attention. However, despite the ability for these modifications to enhance BERT's ability to capture long-range dependencies, they also introduce limitations such as increased computational costs and decreased performance on specific tasks.

In this paper we explore self-supervised attention (SSA) as an alternative approach to enhance BERT's attention mechanism. Inspired by the paper "Improving BERT with Self-Supervised Attention" by Clark et al. (2021), we investigate the impact of SSA on BERT's performance in various tasks. By training the attention mechanism alongside the model using a masked word prediction based on the surrounding context and attention scores, we apply self-supervised learning to the attention mechanism itself.

To assess the effectiveness of SSA we recreate the SSA mechanism, utilize BertViz for visualization to gain insights into the model's behavior, and evaluate our model on multiple datasets, including the Stanford Sentiment Treebank, Quora Paraphrase dataset, SemEval STS dataset, and the SQuAD benchmark. Our findings reveal that SSA improves performance when fine-tuning on a single task, corroborating existing studies showing its potential to capture long-range dependencies more effectively.

However, when fine-tuning on multiple tasks simultaneously, we were not able to find any significant performance benefits due to SSA. This observation highlights the complexity of incorporating self-supervised attention mechanisms in multitask learning scenarios and raises questions about the adaptability of SSA in these settings. It also suggests that alternative attention mechanisms, such as sparse or global attention, might be more suitable for enhancing the model's performance in multitask settings.

This study hopes to contribute to the growing body of research on attention mechanisms in natural language processing by providing a comprehensive evaluation of self-supervised attention in single-task and multitask learning scenarios. Furthermore, by analyzing the model's behavior through BertViz, we show how visualization tools can examine and interpret the attention mechanism, laying

the groundwork for future research on more effective and efficient attention mechanisms for natural language processing tasks.

2 Related Work

Several alternative approaches have been proposed to improve BERT's self-attention mechanism for capturing long-range dependencies. Sparse attention methods such as Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020) introduce sparse connectivity patterns to allow the model to attend to more distant tokens with a lower computational cost. While the traditional self-attention mechanism attends to every token in the input sequence leading to quadratic computational complexity, sparse attention mechanisms focus on a smaller subset of tokens which reduces computational costs and allows for the processing of longer input sequences. However, it's hard to design sparse connectivity patterns that balance efficiency and representational power. This often requires extensive experimentation and may be task-dependent, which can limit the applicability of a specific sparse pattern across various NLP tasks.

The other most common approach is global attention mechanisms, such as in Transformer-XL (Dai et al., 2019) and Compressive Transformer (Rae et al., 2019), which use an attention mechanism that focuses on specific tokens and maintains global context across sequences. Global attention methods are similarly able to reduce computational complexity by selectively attending to a subset of key tokens that provide essential context for the task at hand. However, they suffer from similar limitations as sparse approaches. Focusing primarily on globally relevant tokens causes these methods to also lose some fine-grained local information that could be important for specific tasks or contexts, subsequently leading to reduced performance on tasks where local context or token-level details are crucial. Determining the most globally relevant tokens is also difficult due to it depending on the specific task and input data, and making a sub-optimal selection of global tokens also leads to negative impacts on the model's performance.

This paper explores the idea of self-supervised attention proposed by Clark et al. (2021), which trains the attention mechanism alongside the model by predicting a masked word based on the surrounding context and attention scores. This approach aims to improve BERT's ability to capture long-range dependencies without introducing the limitations of sparse or global attention mechanisms, as the self-supervised aspect of it removes the difficult barriers of designing sparse connectivity patterns and finding the most globally relevant tokens. The original paper didn't clearly articulate what the limitations of self-supervised attention were in contrast to alternate methods, so this paper also seeks to elucidate those limits.

3 Approach

We start by implementing a Multitask BERT architecture with Multi-headed self-attention, but only pretrained and finetuned on sentiment analysis prediction. This provides a baseline for both multitask pretraining and finetuning as well as self-supervised attention.

After pretraining and finetuning the model on sentiment analysis prediction, we then extend the BERT model with self-supervised attention, which trains the attention mechanism alongside the model by predicting a masked word based on the surrounding context and attention scores. To achieve this, we modify the forward methods in the BertModel, BertLayer, and BertSelfAttention modules to extract and return the attention weights. We then add a self-supervised attention loss term to the training loss, encouraging the attention mechanism to focus on relevant context tokens when predicting masked words.

Finally, we employ a Multitask BERT architecture, fine-tuning our model on multiple tasks, including sentiment analysis, paraphrase detection, and semantic textual similarity using the Stanford Sentiment Treebank, Quora Paraphrase dataset, SemEval STS dataset, and also the SQuAD benchmark. The Multitask BERT, Dataset, and Evaluation pipelines are extended to include the question and answer prediction necessary for the SQuAD 2.0 benchmark. This question and answer prediction also has the option of not producing an answer if there isn't sufficient information in the context, which is a new feature of the SQuAD 2.0 benchmark relative to the original. We use the extracted attention weights to visualize the attention mechanism and analyze its effectiveness in capturing long-range dependencies, and compare results to determine efficacy of different model architectures and parameters.

4 Experiments

4.1 Data

We evaluate our model on four datasets:

Stanford Sentiment Treebank (SST) (Socher et al., 2013): A dataset for sentiment analysis, consisting of movie reviews annotated with sentiment labels ranging from very negative to very positive. The task is to predict the sentiment label of a given sentence.

Quora Paraphrase dataset: A dataset containing pairs of questions from Quora, labeled as either paraphrases or non-paraphrases. The task is to predict whether a given pair of questions is a paraphrase or not.

SemEval STS dataset (Agirre et al., 2012; Cer et al., 2017): A dataset for the Semantic Textual Similarity (STS) task, where the goal is to predict the semantic similarity between pairs of sentences on a continuous scale from 0 (completely unrelated) to 5 (semantically equivalent).

SQuAD 2.0 benchmark (Rajpurkar et al., 2018): An extension of the original SQuAD reading comprehension dataset, SQuAD 2.0 combines the original SQuAD dataset with over 50,000 unanswerable questions. The task involves predicting the answer to a given question based on the provided context paragraph, while also determining if the question is unanswerable from the given context.

4.2 Evaluation method

We use the following evaluation metrics for each task:

Sentiment Analysis (SST): We use accuracy as the evaluation metric, which measures the proportion of correctly classified sentiment labels.

Paraphrase Detection (Quora): We use the F1 score and accuracy to evaluate performance. F1 score considers both precision and recall, while accuracy measures the proportion of correctly classified paraphrase pairs.

Semantic Textual Similarity (SemEval STS): We use the Pearson correlation coefficient to measure the correlation between the predicted similarity scores and the ground-truth scores.

SQuAD benchmark: We use the F1 score as the evaluation metric. F1 score measures the overlap between the predicted answer and the ground-truth answer.

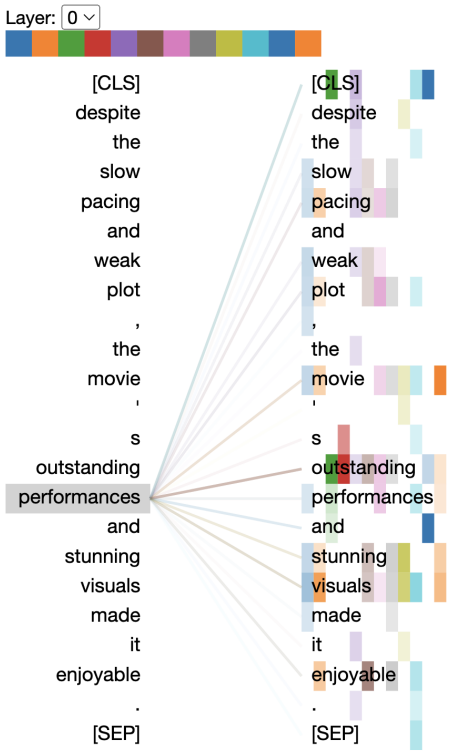
4.3 Experimental details

We use the BERT-base architecture as our starting point, with a hidden size of 768, 12 layers, and 12 attention heads. We first train our model on the SST dataset for 15 epochs with a learning rate of 1e-3, a batch size of 64, and the AdamW optimizer with a weight decay of 0.01. We then finetune on the SST dataset for 15 epochs with a learning rate of 1e-5. Next, we add SSA and repeat the same process as before. Finally, we add the ability to predict a paraphrase, predict similarity, and predict question answers, and we then pretrain and finetune on all tasks for 10 epochs with the same hyperparameters as before. We end by adding SSA loss to every prediction task and training and finetuning again on all tasks. We take the average of the SST, PARA, and STS scores and report them as well.

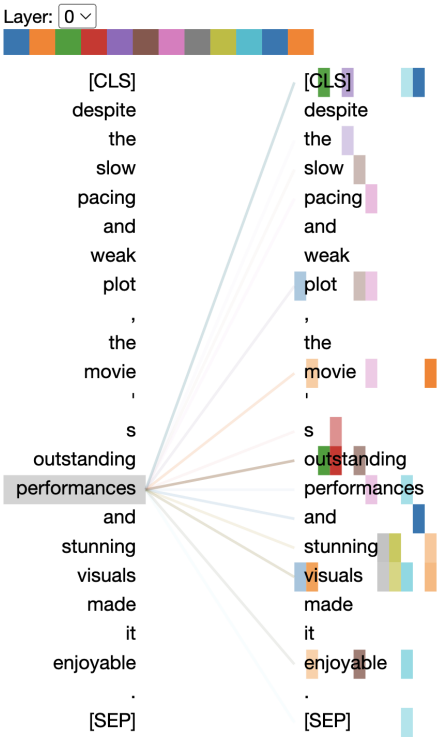
4.4 Results

Model	SST	PARA	STS	SQuAD	Overall
MULTI + P/F + SSA	0.479	0.695	0.315	53.4	0.496
MULTI + P/F - SSA	0.503	0.609	0.288	54.5	0.467
SINGLE (SST) + P/F + SSA	0.540	0.412	0.063	N/A	0.338
SINGLE (SST) + P/F - SSA	0.512	0.421	0.072	N/A	0.335

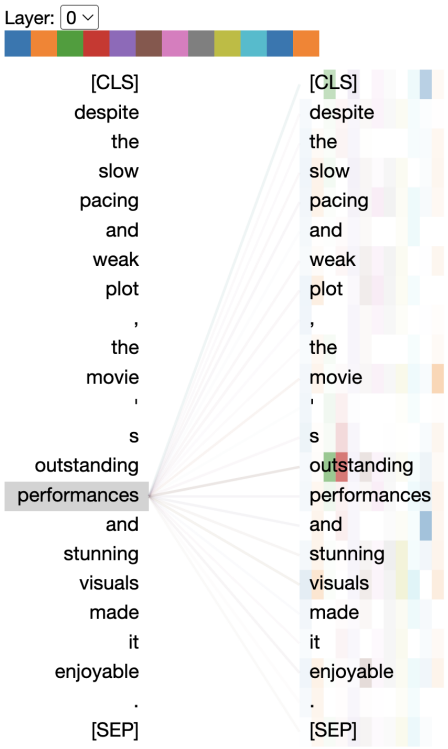
Table 1: Model Performance



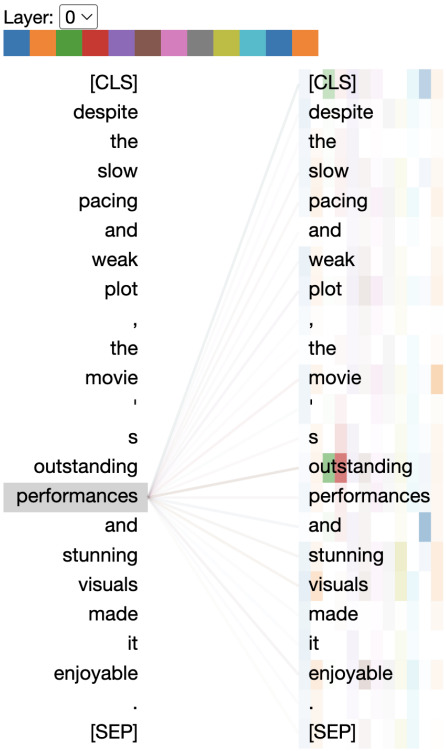
(a) Single-Task, No SSA, 3



(b) Single-Task, SSA, 4



(c) Multi-Task, No SSA, 2



(d) Multi-Task, SSA, 2

Figure 1: caption

4.5 Visualization

To better understand and interpret the role of the self-supervised attention mechanism in the token embeddings we employed BertViz to create self-attention maps. These maps are shown on the previous page, with one map per model. Each map is showing the attention weights for the token "performances" in the sentence "despite the slow pacing and the weak plot, the movie's outstanding performances and stunning visuals made it enjoyable," which is a sentiment analysis problem that requires the ability to capture long-range dependencies to balance out the negative sentiment of the initial part of the review. The predicted sentiment is provided after the model name under each map, with single-task pretraining and finetuning combined with self-supervised attention predicting the highest sentiment for the example.

5 Analysis

Our findings reveal that supervised self-attention (SSA) leads to improved performance when fine-tuning on a single task. However, when fine-tuning on multiple tasks, the benefits of SSA become less significant. To better understand this phenomenon, we examine the attention maps generated during the experiments. For multitask models, the attention maps display weak and ambiguous connectivity patterns, whereas single-task pretraining shows more definitive and robust connections.

When training with SSA on a single task, the model demonstrates a more focused attention mechanism, as evidenced by the fewer but stronger connections in the attention maps. This can be attributed to the self-supervised learning process, which promotes a more selective and efficient utilization of attention resources to capture relevant information within the given context. The improved performance on the single task suggests that SSA effectively guides the model to better capture long-range dependencies and complex semantic relationships.

However, in the multitask setting, the lack of a noticeable impact on the attention maps implies that the advantages of SSA are not as pronounced. The increased complexity of learning multiple tasks simultaneously may dilute the effect of the self-supervised attention mechanism. As the model needs to adapt its attention patterns to accommodate a diverse set of tasks, the specialization afforded by SSA in single-task scenarios becomes less beneficial. The absence of a significant performance improvement in multitask learning highlights the challenges of integrating self-supervised attention mechanisms in such settings.

This analysis underscores the difficulties of incorporating self-supervised attention mechanisms in multitask learning and suggests that alternative attention mechanisms, such as sparse or global attention, might be more appropriate for enhancing the model's performance in multitask settings. The limitations of the current study also indicate the need for further investigation into the interplay between self-supervised attention and multitask learning. Exploring alternative model architectures, optimization strategies, and task-specific attention mechanisms can provide deeper insights into the dynamics of self-attention in multitask learning scenarios and help identify more effective approaches for improving performance across diverse tasks.

6 Conclusion

In this work, we explored a novel approach to enhance BERT's attention mechanism by incorporating self-supervised attention (SSA). Our method demonstrates improved performance when pretraining and finetuning on a specific task such as sentiment analysis, all without significantly increasing computational costs. We found that our model is better equipped to capture long-range dependencies due to its training process of predicting masked words using faraway tokens, contributing to its enhanced performance over the original BERT model which does not have an explicit incentive to attend to distant tokens.

The main takeaway from our exploration is that the benefits of SSA become less pronounced when fine-tuning on multiple tasks, suggesting that the multitask learning scenario presents additional challenges for self-supervised attention mechanisms. This phenomenon can be attributed to the increased complexity of learning multiple tasks simultaneously, which may dilute the effect of the self-supervised attention mechanism. As the model needs to adapt its attention patterns to accommodate a diverse set of tasks, the specialization afforded by SSA in single-task scenarios

becomes less beneficial. Furthermore, the addition of an extra task-specific training mechanism for every task being trained on causes a significant increase in computation which means it takes longer to get to reasonable scores when you have more tasks, so the impact is likely more diluted.

This project has some limitations. A more thorough search for optimal hyperparameters should be done for the multitask model as this study only used the best hyperparameters from the single task model due to the excessive training time associated with a multitask model grid search. Given more time, the multitask model could be run for more epochs and with different parameters including the SSA weight parameter, which potentially could lead to improved performance. However, it's worth noting that one epoch takes 40 minutes on an A100 GPU, so lots of compute power will be necessary to do so.

Future work may explore other forms of self-supervised learning to further improve the attention mechanism or investigate hybrid attention mechanisms that combine the benefits of sparse, global, and self-supervised attention. These alternative mechanisms could potentially address the specific issues presented by self-supervised attention in multitask learning settings by offering a more adaptable attention mechanism that caters to the diverse requirements of multiple tasks without suffering from the same shortcomings as SSA.