BERT and MNRLLie: Extending minBERT with Deep Metric Learning and Gradient Surgery

Stanford CS224N Default Project

Ben Auslin
Department of Computer Science
Stanford University
bauslin@stanford.edu

Jorge Martinez-Alba
Department of Computer Science
Stanford University
jorgema@stanford.edu

Henry Bradley
Department of Computer Science
Stanford University
henryab@stanford.edu

Abstract

In this project, we implemented key aspects of the original BERT model, such as multi-head self-attention and a Transformer layer, to create a functional BERT model from scratch. We utilized this model to perform sentiment analysis on two different datasets: the Stanford Sentiment Treebank dataset and a dataset of movie reviews. In the latter half of the project, we fine-tuned and extended the BERT model to create sentence embeddings that can perform well across a wide range of downstream tasks. Overall, our project provides a comprehensive understanding of the BERT model and its capabilities in NLP tasks. Our implementation and fine-tuning of the BERT model provide insights into the effectiveness of the model in various downstream tasks.

1 Key Information to include

- Mentor: Cathy Yang
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Finetuning pretrained language models for multiple tasks presents an especially diffcult problem of representation. The usual questions surrounding network architecture, loss functions, and evaluation metrics become compounded by interactions between different tasks and the objective of holistically evaluating performance. Our project explores these issues through extensions of the MinBERT architecture for three sentence-oriented tasks: a discrete five-point scale sentiment classification on the Stanford Sentiment Treebank (SST) dataset, a binary classification for sentence pairs in the Quora paraphrase detection dataset, and a five-point scale similarity classification on the SemEval Semantic Textual Similarity (STS) dataset.

Our experiments were focused on investigating the impact of different finetuning strategies, network architectures, and multi-task learning approaches on the performance of the extended MinBERT model. We evaluated several loss functions and their combinations to optimize the performance on each task while also considering the trade-offs between task-specific and multi-task learning.

3 Related Work

The main goal of finetuning for the tasks in this project was to induce embeddings from BERT specifically to capture the meaning of whole input sentences. Our earlier experiments implemented ideas from Reimers and Gurevych (2019) who demonstrate finetuning BERT as a siamese network produce semantically rich full-sentence embeddings. A siamese network is an application of the same model to two inputs, yielding two embeddings which are then measured for similarity. The similarity measurement is combined with a true label for the pair in an objective function specific to the type of training task.

The paper also presents cosine-similarity, defined as

$$\text{Similarity}(w_i, w_j) = \frac{w_i \cdot w_j}{\|w_i\|_2 \|w_j\|_2}$$

combined with mean-square error loss as a regression objective function. Although the paper also presents a classification objective function, we opted for a different loss framework for both the single-sentence sentiment and paired-sentence paraphrase classification tasks.

To explore beyond the approaches of Reimers and Gurevych (2019), we looked at deep metric learning, which is an approach that aims to represent similarity and dissimilarity between using *space* as defined by a learned distance function (Kaya and Bilge, 2019). In deep metric learning, the distance function optimal for the given data and task is learned by a neural network. In our case, taking inspiration from earlier work with LSTMs by Mueller and Thyagarajan (2016), we attempted a simple ℓ_1 loss applied to a per-task final linear projection layer on the pooled transformer output.

One of the primary challenges with multitask learning is that the parameter updates for different tasks may conflict. Yu et al. (2020) address this by adjusting the gradient of each task k with respect to some parameter θ as

$$\mathbf{g}'_{k,t} = \begin{cases} \mathcal{L}_k(\theta) & t = 0 \\ \mathbf{g}'_{k,t-1} - d_t & t \notin \{0, k\} \\ \mathbf{g}'_{k,t-1} & t = k \end{cases}, \quad d_t = \begin{cases} \frac{\nabla_\theta \mathcal{L}_k(\theta) \cdot \nabla_\theta \mathcal{L}_t(\theta)}{\|\nabla_\theta \mathcal{L}_t(\theta)\|_2^2} & \nabla_\theta \mathcal{L}_k(\theta) \cdot \nabla_\theta \mathcal{L}_t(\theta) < 0 \\ 0 & \nabla_\theta \mathcal{L}_k(\theta) \cdot \nabla_\theta \mathcal{L}_t(\theta) \geq 0 \end{cases},$$

where $t \in \{1, \ldots, T\}$ enumerates the tasks in random order, $\mathcal{L}_i(\theta)$ is the loss function of task i and $\mathbf{g}'_{k,t}$ is the adjusted gradient for task k at step t. The iterative procedure described in the formula adjusts conflicting gradients, which are those that share a negative inner product, by orthogonalizing them against each other. The $\mathbf{g}'_{i,T}$ for all i are then summed together to obtain the modified update for θ from all tasks.

4 Approach

Over the course of this project, we used two networks with a similar architecture operated in different modes. The base architecture is the twelve layers of the BERT transformer layer followed by a final linear layer for each of the three tasks. The final linear layer takes the pooled representation from the last transformer layer and outputs logits for each task. In the case of SST, the output is a softmax over the logits for the five sentiment classes. In the case of paraphrase detection, the output is a single logit. STS classification is treated as a regression task because the training data contains non-integral labels, and the output is also a single logit.

Our initial experiments followed Reimers and Gurevych (2019) by running the base architecture as a siamese network. Logically, a siamese architecture contains two identical networks which each process one of the two inputs and whose weights are always the same. In terms of implementation, this is accomplished by sending the two sentences through the forward pass, and accumulating gradients over the two sentences. The two outputs are combined by a function specific to the task and a loss is calculated over the combined result. We found limited success with this mode of operation. A selection of results using our siamese network on a subset of the training data can be found in the ablation study below.

Following the siamese network, we encoded sentence pairs using the separator token as originally described in Devlin et al. (2018). We refer to this below as the *segments* network because of the learned segment embedding also described in the same paper. The segments are the spans of the input

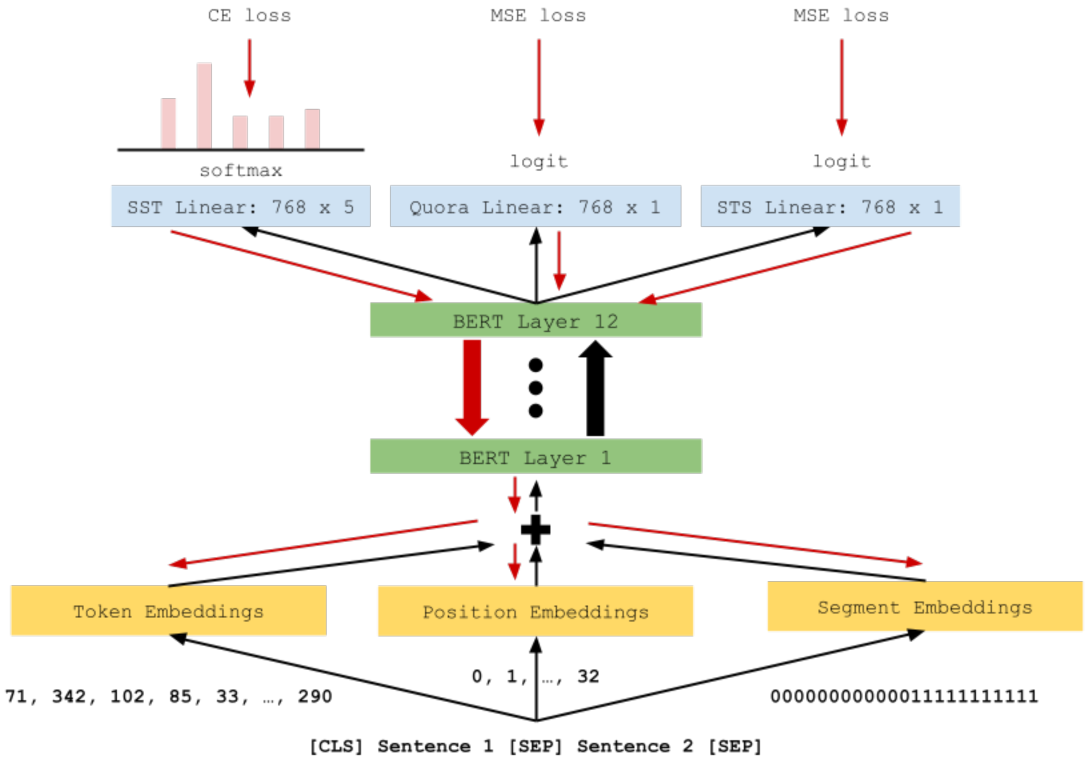

Figure 1: Final network architecture using input segments

sequence corresponding to the two sentences and are represented as a binary mask when provided as input to the embedding layer. Sending a pair of sentences encoded together through the network together increased performance by a large margin. Our final network and leaderboard scores use input segments. Further analysis is provided in later sections.

Our training loop trains on one batch from each task in each iteration. To deal with the length of the Quora dataset, both shorter training sets were repeatedly sampled, restarting the epoch when necessary, until the entire Quora training set was consumed. The intent of this strategy was to avoid overweighting the paraphrase detection training in each epoch. To mitigate conflicting parameter updates when training multiple tasks, our final network was trained using gradient surgery as detailed in the previous section. An AdamW update step was calculated from the result of gradient surgery.

5 Experiments

5.1 Data

We used the given datasets for the default final project, being Stanford Sentiment Treebank (SST), which contains 11,585 sentences from movie reviews and CFIDMB, which contains 2,434 movie reviews. These datasets are pre-split into training, dev, and test sets and were used for our initial sentiment classification task.

For the extensions, we used the given Quora and SemEval STS Benchmark datasets, for sentiment analysis, paraphrase detection, and semantic textual similarity, respectively. The Quora dataset has 400,000 question pairs, while SemEval has 8,628 sentence pairs. These datasets are also pre-split into training, dev, and test sets.

5.2 Evaluation method

We evaluated against the baseline BERT scores shown in the handout:

1. Pretraining for SST: Dev Accuracy: 0.390 (0.007)
2. Pretraining for CFIMDB: Dev Accuracy: 0.780 (0.002)
3. Finetuning for SST: Dev Accuracy: 0.515 (0.004)
4. Finetuning for CFIMDB: Dev Accuracy: 0.966 (0.007)

For our midpoint model, we did not do exceptionally well relative to the leaderboard, achieving the following scores:

1. Pretrain SST Dev Accuracy: 0.378
2. Pretrain CFIMDB Dev Accuracy: 0.731
3. Finetuning SST Dev Accuracy: 0.516
4. Finetuning for CFIMDB Dev Accuracy: 0.967

For the multitask model, performance on the SST sentiment analysis and Quora paraphrase detection tasks are evaluated in terms of accuracy. Pearson's correlation coefficient is used to measure regression performance on the STS dataset. These measurements are reported in the tables of the results section below.

5.3 Experimental details

In our ablation study, we used the first 2048 examples in each training set, the full development set for evaluation, a batch size of 32 in each task, 6 epochs, and an initial learning rate of 3×10^{-5}. The initial learning rate is of less importance because all experiments optimized with momentum using AdamW. For the submitted scores, the same setup was used with the change that 4 epochs, as described in section 4, over the entire training set were completed.

Further details on the ablation study are as follows. The siamese and segments networks both computed the softmax over the five classes of the SST and used cross-entropy loss. The siamese network output $d = \|u - v\|_2$, where u and v are the two mean-pooled network outputs, for both Quora paraphrase detection and STS regression. On the Quora dataset, the siamese network applied contrastive learning loss, defined as

$$\frac{1}{2}(1-y)d^2 + \frac{1}{2}y(\max(0, m-d))^2$$

in Kaya and Bilge (2019), and mean squared error (MSE) loss for STS. The margin m was set to 0.5. The segments network output the sigmoid activation of its final linear layer on the Quora dataset and also applied contrastive loss. For the STS dataset, the segments network produced a plain logit and computed MSE loss.

In an additional experiment, we wanted to see the effectiveness of Multiple Negatives Ranking Loss, given by

$$\mathcal{J}(x, y, \theta) = -\frac{1}{K}\sum_{i=1}^{K}\left[S(x_i, y_i) - \log\sum_{j=1}^{K} e^{S(x_i, y_i)}\right]$$

Henderson et al. (2017) For our implementation, we used Cosine Similarity for our scoring function.

Unfortunately, due to our time constraints and limited computing capacity by the end of the project, we were unable to get results in conjunction with the rest of our experimental parts. In isolated testing, we were able to use our Multiple Ranking Loss on our Quora and STS datasets where we compared the results of paraphrasing and similarity.

The submitted scores use the segments network with the final sigmoid activation removed for paraphrase detection. As part of training, the sigmoid is applied to the logit before computing MSE loss.

5.4 Results

Leaderboard test scores:

- SST test Accuracy: 0.500
- Paraphrase test Accuracy: 0.720
- STS test Correlation: 0.725
- Overall test score: 0.649

Ablation study: The following table reports results for the full development set. "GS" indicates that the gradients due to each task were computed separately and cached, then gradient surgery was applied to all cached gradients before passing to AdamW and applying the update. "no GS" indicates an AdamW update step was applied after each batch of each task on unaltered gradients. All configurations use mean pooling on the final hidden states, except for one experiment that tested downstream tasks on the final CLS token state. One experiment tested ℓ_1 loss instead of MSE for both paraphrase detection and semantic textual similarity. Two additional experiments randomized the order in which batches from each task were sent through the network in each iteration.

Configuration	SST	Quora	STS
siamese, no GS	0.254	0.375	0.021
siamese, GS	0.253	0.375	-0.105
segments, no GS	0.278	0.375	0.743
segments, GS, mean pooling	**0.282**	0.375	**0.765**
segments, GS, CLS pooling	0.274	0.375	0.726
segments, GS, ℓ_1 loss	0.280	0.375	0.760
segments, no GS, rand task order	0.278	0.375	0.728
segments, GS, rand task order	0.268	0.375	0.751

Earlier in our project, while developing our multitask pipeline, we trained the siamese network for four epochs over the full training set using cosine similarity and MSE loss as originally described in Reimers and Gurevych (2019) for paraphrase detection and semantic similarity. After switching to contrastive loss, the performance on the two sentence pair tasks decreased when the siamese network was trained on the same data again for four epochs. The following table also reports results for the full development set.

Configuration	SST	Quora	STS
siamese, GS, cosine sim, MSE loss	0.470	0.471	0.042
siamese, GS, cosine sim, contrastive loss	0.470	0.375	0.014

We note that using contrastive loss appears to limit accuracy on the paraphrase detection task at 0.375 as shown in the ablation study above. Taking this into account, we trained our final model using MSE loss on both sentence pair tasks.

In our isolated experiment with Multiple Negatives Ranking Loss, we found that our loss function had comparable accuracy to our segments, GS, and mean pooling configuration for our STS dataset. Given more time, if we had factored in Multiple Negatives Ranking Loss into our other configurations there may have been an increase in our STS results. Multiple Negatives Ranking Loss didn't really have noticeable increase from MSE loss in our Quora dataset.

6 Analysis

As demonstrated by the ablation study, the siamese network architecture did not work at all for semantic textual similarity. The paired-sentence encoding with segment embeddings from the original BERT paper resulted in a dramatic improvement in correlation. We hypothesize this is because the original BERT-base network has been trained on pairs of sentences using segment embeddings, so finetuning with the same encoding utilizes language knowledge stored in the pretrained weights. In contrast, finetuning the model as a siamese network introduces a mode of operation that the model initially has no knowledge of. Given our limited training time and data, it is reasonable that instead of trying to retrain all the layers underneath to work with two forward passes, continuing to use the input structure of the pretrained model worked best.

Although Mueller and Thyagarajan (2016) report that an ℓ_1 distance measurement on siamese network outputs outperformed competing models even on benchmarks calibrated for MSE, our network experienced reduced performance when using both ℓ_1 loss. We have two possible explanations for the discrepancy. First, our ℓ_1 experiment was done with input segments to encode sentence pairs, so there was no representation space for individual sentences in which an ℓ_1 could be computed. We instead used an ℓ_1 measurement between the labels and logits. The output of the siamese network in Mueller and Thyagarajan (2016), however, provides a decoupled representation for both input sentences. A second explanation is that our underlying network architecture uses contextual representations that are learned during training. This differs significantly from the pretrained word2vec embeddings employed by Mueller and Thyagarajan (2016).

Surprisingly, gradient surgery had less impact than expected. In the case of the segments network, the impact of gradient surgery was less than changing the pooled output representation. We suspect this is because there only three tasks. In particular, it is possible that the tasks required similar information from the network and so had limited gradient conflicts. As evidence of this, we note that in both the ablation study and submitted results, the segments network had far better performance on the two paired-sentence tasks than the single sentence classification. The paired-sentence tasks are similar in nature, comparing two sentences for meaning, and so those tasks seem to have experienced constructive interference in their training signals. The unrelated single-sentence sentiment classification seems to have experienced a diminished training signal, which would happen if its gradient were to lose magnitude when subtracting out components antiparallel to the other two tasks during gradient surgery.

We found that our Multiple Negatives Ranking Loss performed well with the STS dataset. This is due to how Multiple Negatives Ranking Loss handles similarity. We bring the similar sentences closer to one another, while we create distance between all of the sentences that are not marked as similar. This strategy resulted in good similarity classification.

We also attempted to implement a named entity recognition task, but there was not sufficient time to complete a version that worked with the rest of the model. Given more time, we believe that the model would have performed at a level comparable to similar models.

7 Conclusion

Our main finding is that using paired sentences with segment embeddings far outperformed finetuning with two forward passes. Phrasing the lesson from our analysis above more concisely, the power of finetuning lies in its use of the general aptitude for language imparted on the network during pretraining. Changes to the input or operation of the network operates can lead to the loss of language ability.

Regarding prior work with deep metric learning, we found that ℓ_1 distance learning led to less performance than comparable methods when applied in our multitasked finetuning setting. We hypothesized that this was due to the lack of a representation space for individual sentences when sending pairs of sentences through the network together. We suspect that other techniques from the literature, particularly those developed using traditional RNN architectures, would also experience discrepancies in effectiveness when ported to the transformer architecture.

For ℓ_1 distance learning and our other negative results, multitask training was likely a confounding factor since we were unable to isolate one task to analyze performance discrepancies. Further study of these negative results and their failure modes can be attempted by isolating individual tasks. A similar remark can be made of gradient surgery. In our setting with limited multitasking, gradient surgery did not have too noticeable of an effect. A more extensive selection of tasks, especially those known to produce conflicting updates, would provide better data on the impact of gradient surgery. Finally, keeping other parts of the system such as input encoding, output values, and loss functions constant would allow a better study of how techniques may differ in effectiveness across network architectures.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Matthew L. Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *CoRR*, abs/1705.00652.

Mahmut Kaya and H. Bilge. 2019. Deep metric learning: A survey. *Symmetry*, 11:1066.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

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.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc.