Contrastive Pretraining of minBERT to Improve Performance in Downstream Tasks

Stanford CS224N Default Project

Nick Phillips
Department of Computer Science
Stanford University
nphill22@stanford.edu

Abstract

Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based model that generates contextual word representations. By extending BERT, these learned representations of language can be applied across a range of tasks. In this project, we evaluate the performance of a BERT-based multitask model in sentiment analysis, paraphrase detection, and semantic textual similarity, using both pretrained and fine-tuned weights. We then evaluate the effects of contrastive pretraining on model performance using the SimCSE framework. We demonstrate that pretraining using contrastive learning objectives yields sentence embeddings that outperform baseline methods after task-specific fine-tuning. Finally, we perform an error analysis to qualitatively characterize the effects of contrastive pretraining and the limitations of our model.

1 Introduction

Learning contextual word representations is a central problem in natural language processing (NLP) with innumerable downstream applications. For representations to be useful, they should capture the relevant semantic and syntactic meanings of words in their context. Therefore, it is sensible to derive meaningful representations of words as a function of surrounding words. Methods such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) apply this principle to learn vector representations by either maximizing the probability of observing a word given its context or maximizing the probability of observing the context given the word. While this class of methods has proven to be broadly applicable, such representations often fail to accurately encode the nuanced meanings of language.

Applying more expressive models to the problem of learning word representations dramatically improves our ability to contexually model language. Neural network architectures, such as recurrent neural networks (RNNs) and transformers (Vaswani et al., 2017), have enabled learning word representations that more accurately capture details such as polysemy, word ordering, and complex dependencies in language. These representations can be applied to more accurately reflect the meaning of entire sentences in which the words were used. For example, representations can be used to determine if a sentence conveys a positive or negative sentiment. Representations from pairs of sentences can be used to determine how similar two sentences are to one another, or if one sentence paraphrases the other. In the project, we implement and extend a transformer-based model known as Bidirectional Encoder Representations from Transformers (BERT) to perform these tasks.

As an encoder model, BERT generates representations by passing token embeddings through successive transformer blocks, comprising multi-headed attention, residual connections, layer normalization, and feed-forward layers. While this architecture has recently achieved state-of-the-art performance in various NLP evaluations, the model is relatively large and therefore require significant data and resources to train from random initialization. To mitigate this, we utilize model weights that were pretrained on unsupervised masked token prediction and next sentence prediction. The pretrained

BERT encoder model forms the base of our architecture, and we extend this model by attaching task-specific heads with learnable parameters to perform sentiment analysis (SST), paraphrase detection (PARA), and semantic textual similarity (STS).

Additionally, we extend our analysis to explore the impact of contrastive pretraining on model performance in these tasks. We perform this training using the SimCSE framework with both supervised and unsupervised contrastive objectives. We then fine-tune our contrastive pretrained model on the sentiment analysis, paraphrase detection, and textual similarity tasks. We conclude by performing an error analysis, where we qualitatively characterize the impact of contrastive pretraining and identify limitations of our model.

2 Related Work

2.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a language representation model that applies the transformer encoder architecture to produce contextual representations of input tokens (Devlin et al., 2019). These representations are trained using unsupervised objectives, such as masked token prediction and next sentence prediction. In this paper, the authors demonstrate that a large pretrained BERT model can be extended with additional layers and task-specific fine-tuning to achieve state-of-the-art performance in a wide range of tasks. This was a transformative publication in NLP, and BERT became the basis for many high-performance language models. In this project, we leverage a variant of the BERT model to produce sentence embeddings for our downstream tasks.

2.2 SimCSE

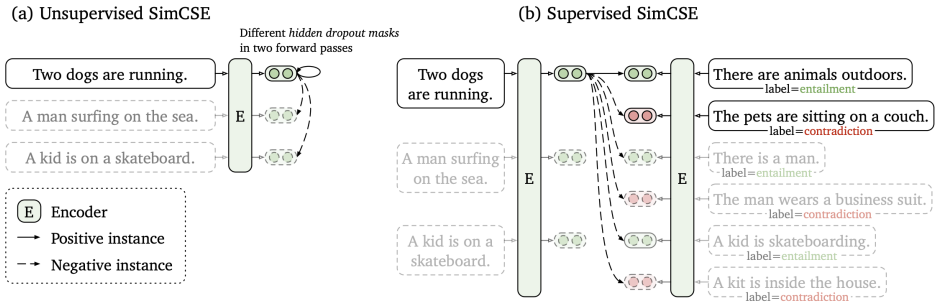

Figure 1: Graphical overview of the SimCSE contrastive objectives.

SimCSE is a contrastive learning framework that aims to improve the quality of sentence embeddings by fine-tuning on a supervised or unsupervised objective (Gao et al., 2022). Two key concepts in contrastive learning are positive pairs, which are semantically related entities, and negative pairs, which are unrelated entities. Training using the SimCSE contrastive objectives attempts to maximize the similarity of encodings for positive pairs of data while minimizing the similarity of negative pairs. In the unsupervised context, a positive pair comprises two embeddings of the same input with different dropout masks, essentially employing dropout as a data augumentation strategy. In the supervised context, a positive pair comprises two sentences that are entailments of one another. In both contexts, negative pairs are taken as all other combinations of sentences in a training batch.

The authors train with a contrastive cross entropy loss, and demonstrate that the learned embeddings maintain alignment and improve uniformity, producing more isotropic representations. They demonstrate that fine-tuning with contrastive embeddings improves performance in various tasks over BERT and RoBERTa baselines. The SimCSE framework is the basis of our contrastive objectives in this project.

3 Approach

3.1 minBERT

We begin with model design and implementation. Our multitask model is based on minBERT, a minimal implementation of BERT. We manually implemented the embedding layer and the transformer block comprising multi-head attention, layer normalization with residual connections, and a feed-forward sublayer. We also implemented the Adam optimizer with weight decay, which we used exclusively in our analysis (Kingma and Ba, 2017). Finally, we downloaded weights for minBERT that were pretrained using unsupervised masked token prediction and next sentence prediction.

Our BERT model has the following configuration: We use a vocabulary size of 30,522, a model dimensionality of 768, learnable token embeddings, and absolute positional embeddings. Our model is comprised of 12 transfomer blocks, each using multi-head attention with 12 attention heads and a 3072 dimensional intermediate hidden size with GELU activation. We apply dropout to both attention scores and before layer normalization, with a dropout probability of .3.

3.2 MulticlassBERT

We then extended our minBERT model for multiclass prediction. Sentence embeddings are generated by performing a forward pass and extracting the pooler output for the [CLS] token. With these embeddings, sentiment prediction is performed by applying dropout and a single linear layer of dimension $[hidden\_size, 5]$. For paraphrase prediction, we generate embeddings for each sentence using the previously described method, concatenate the embeddings, and apply dropout and a single linear layer of dimension $[2 * hidden\_size, 1]$. Finally, for similarity prediction, we generate embeddings for both sentences, compute cosine similarity, and pass the similarities into a ReLU activation function to restrict to positive values. We then scale the activations by a factor of 5 to reflect the range of similarity labels.

3.3 Multiclass Baselines

We evaluate two baseline approaches starting from pretrained minBERT weights: 1) training only task-specific model heads, and 2) fine-tuning all model weights. For our task-specific training, we freeze all model weights for the BERT base model, and only train the linear layers on top of BERT sentence embeddings. We note that because the similarity output contains no learnable parameters, in this baseline the similarity values reflect only the baseline BERT sentence embeddings. In the fine-tuning baseline, all model weights for the BERT model and task-specific heads are trained.

In multi-class training, we employ a per-batch, round-robin training strategy. For each iteration, we take one batch of training data from our datasets for SST, PARA, and STS. We then compute loss for each of the tasks individually and perform a step with our optimizer. Because the datasets are of variable length, we define a training epoch as one epoch for our smallest dataset. Therefore, in a single training epoch, larger datasets will train using only a fraction of the available data. For both SST and PARA we use cross entropy loss, and for STS we use mean squared error loss.

For each baseline model, we evaluate performance across all tasks on the development set. For both sentiment analysis and paraphrase prediction we report performance as accuracy; on semantic textual similarity we report Pearson correlation. We also report performance on the test set for our best performing model.

3.4 Contrastive Pretraining

We then extend our training strategy with contrastive pretraining. We apply the SimCSE framework, and train using both contrastive objectives: an unsupervised task of predicting input sentence from itself, and a supervised task of identifying entailment or contradiction. Contrastive training uses the following cross-entropy objective: for representations $\mathbf{h}_i$ of x_i and $\mathbf{h}_i^+$ of x_i^+, where (x_i, x_i^+) are a semantically related pair in a minibatch of size N, with temperature parameter τ,

$$\ell_i = -\log \frac{e^{sim(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}} \qquad sim(\mathbf{h}_1, \mathbf{h}_2) = \frac{\mathbf{h}_1^\top \mathbf{h}_2}{||\mathbf{h}_1|| \cdot ||\mathbf{h}_2||} \tag{1}$$

In the unsupervised case, the positive pairs are generated by two forward passes of the network with dropout. We perform unsupervised pretraining using text from all of the available datasets. In the supervised case, positive pairs are taken as valid paraphrases or sentences with a similarity score of 4 or 5, with all other sentences in batch as negatives. Following pretraining with this objective, we perform fine-tuning and evaluation as previously described in our baseline methods. We manually implemented the contrastive loss and training procedure.

4 Experiments

4.1 Data

Some key attributes of our datasets are shown below:

	Stanford Sentiment Treebank	Quora Dataset	SemEval STS Dataset
Train Size	8,544	141,506	6,041
Label	Categorical [0,1,2,3,4]	Binary	Integer [1 to 5]
Task	Sentiment Prediction	Paraphrase Detection	Semantic Textual Similarity

4.1.1 Stanford Sentiment Treebank

The Stanford Sentiment Treebank consists of 11,855 sentences from movie reviews (Socher et al.). Each phrase was annotated by 3 human judges to label the sentence as [negative, somewhat negative, neutral, somewhat positive, or positive], which corresponds to numeric labels [0, 1, 2, 3, 4]. The dataset is split with 8,544 sentences in the training set, 1,101 sentences in the dev set, and 2,211 in the test set. This dataset is used to train the SST task.

4.1.2 Quora Dataset

The Quora dataset consists of 177,149 pairs of questions from the website Quora with binary labels to indicate if the questions are paraphrases of each other. The dataset is split with 114,506 pairs in the training dataset, 22,212 pairs in the dev dataset, and 40,431 pairs in the test dataset. This dataset is used to train the PARA task.

4.1.3 SemEval STS Dataset

The Semantic Textual Similarity dataset consists of pairs of sentences from text, labeled from 0 to 5 to indicate degree of similarity (Agirre et al., 2013). A 5 indicates identical similarity, while a 0 indicates the sentences are not related at all. The dataset is split with 6,041 pairs in the training dataset, 863 pairs in the dev dataset, and 1,725 in the test dataset. This dataset is used to train the STS task.

4.2 Evaluation method

Performance in the sentiment analysis (SST) and paraphrase detection (PARA) task was evaluated using accuracy. Performance in the semantic textual similarity (STS) task was evaluated using the Pearson correlation of true similarity against the predicted similarity. Evaluation metrics are reported on the development dataset split, and performance of the top model is reported on the test set.

4.3 Experimental details

4.3.1 Baseline Experiments

Baseline models were trained for 10 epochs with learning rate 1e-3 for pretrained weights and 1e-5 for model finetuning. Dropout was set to .3, batch size was set to 8, and the default parameters for the Adam optimizer were used during training. Training using pretrained weights completed in 16.36 minutes, and training using multitask finetuning completed in 48.09 minutes. Both models were trained on a desktop workstation with a NVIDIA 2070 super GPU.

4.3.2 Contrastive Pretraining

Pretraining with contrastive learning objectives were performed for 10 epochs using learning rate 1e-3 for both supervised and unsupervised objectives. Dropout was set to .3, batch size was set to 8, and the default parameters for the Adam optimizer were used during training. Training using the unsupervised objective completed in 19.29 minutes, and training using the supervised objective completed in 20.24 minutes. Both models were then fine-tuned on the multi-task objectives, using the same experimental configuration as the baseline experiments. Fine tuning both the supervised and unsupervised models completed in approximately 51.5 minutes. All models were trained on a desktop workstation with a NVIDIA 2070 super GPU.

4.4 Results

	Dev				**Test**
	Baseline		Contrastive		Contrastive
	Pretrain	Finetune	Unsupervised	Supervised	Supervised
SST (Accuracy)	0.374	0.490	0.492	**0.520**	0.509
PARA (Accuracy)	0.670	0.738	0.740	**0.746**	0.743
STS (Correlation)	0.018	0.670	0.730	**0.738**	0.738
Overall Score (Test)					**0.663**

Table 1: Comparison of baseline and contrastive methods

Results from our analysis are shown in the table above. For our baseline models, we observed that finetuning achieved superior performance to simply training task-specific heads using frozen weights from the baseline BERT model. This effect is most pronounced in the STS task, where the pretrain baseline is almost entirely uncorrelated yet finetuning achieves a correlation of 0.67. This is expected, as the similarity output has no trainable weights, and simply computes the scaled cosine similarity between sentence embeddings; being unable to update sentence embeddings in the pretrain baseline effectively prevents this task from improving throughout training. The dev set metrics in SST and PARA tasks are more similar, although the finetuned model outperforms, potentially due to learning more robust sentence embeddings due to training on the STS task.

Multitask finetuning of the contrastive pretrained model weights improved performance over both the pretrained and finetuned baseline models. Pretraining using the supervised contrastive objective slightly outperformed pretraining using the unsupervised contrastive objective. The task with the greatest magnitude of improvement from pretraining is STS, with a .068 accuracy improvement over the finetuned baseline. This observation makes sense, as the contrastive objective is more similar to the STS objective: maximizing similarity of positive pairs using labels from the paraphrase and similarity datasets essentially pretrains for STS on an expanded dataset.

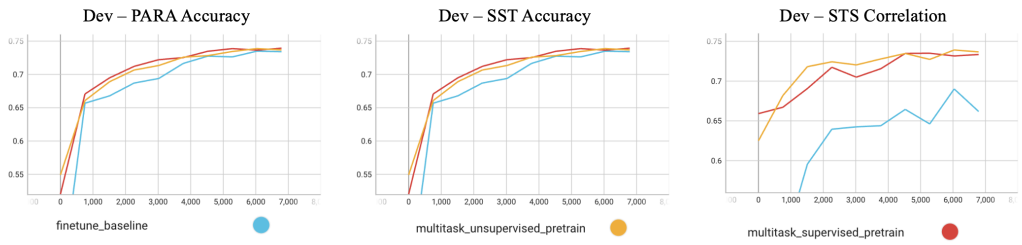

Figure 2: Evaluation metrics on the dev set during multitask finetuning

Contrastive pretraining improved performance in the SST and PARA tasks as well, although by smaller magnitudes. This may be because although the SST and PARA tasks benefit from more robust sentence emebeddings, they are more dependent on training data and task-specific model architecture. As such, finetuning on SST and PARA from both base BERT weights and contrastively trained weights may be similarly upper-bounded in performance with this training configuration and architecture. However, we observed that both supervised and unsupervised pretraining outperformed

the finetuned model in earlier epochs (Figure 2), demonstrating the advantages of starting training from a model that generated better sentences embeddings.

5 Analysis

5.1 Semantic Texual Similarity

We performed an error analysis to identify examples where contrastive pretraining outperformed the finetune baseline and examples where our best model still fails. We initially focused on the STS task, as we observed the greatest improvement in this task. We selected four random sentence pairs where the difference between baseline similarity and true similarity was greater than 2.5.

ID	Sentences	FT	SUP	Label
1	Some ancient historical precedent exists for preferring 10, ... This is indeed possible, but I haven't seen it done experimentally ...	3.63	1.73	0.0
2	To reach John A. Dvorak, who covers Kansas, call, ... To reach Brad Cooper, Johnson County municipal reporter, call ...	4.67	4.28	1.0
3	non-proliferation expert at the international institute for ... senior fellow at the international institute for strategic ...	0.25	0.00	4.0
4	Live Blog: Ukraine In Crisis Live Blog: Iraq In Turmoil	4.54	4.03	0.0

Table 2: Comparison of finetune baseline (FT) and supervised contrastive (SUP) similarity predictions

For comparison 1, the label correctly suggests that the sentences are indeed unrelated. The pretrain baseline predicts a higher similarity than the supervised contrastive model, indicating that contrastive pretraining produced dissimilar sentence embeddings, as desired. For comparison 2, the sentences are extremely related, yet the label suggests that they are dissimilar. In this case, both models predicted similarity above 4, which seems like a reasonable label. Here, it appears that the model is correct and the label is erroneous.

In comparison 3, the label is highly similar, yet both baseline and contrastive models predict extremely low similarity. In this example, both models failed to identify a real similarity. It is difficult to ascertain why the models failed here, but it is worth noting that the sentences contain several proper nouns, acronyms, and unusual diction, which may increase the difficulty of similarity prediction. Finally, in comparison 4, the sentences are very similar yet the label is dissimilar. Both models correctly identify the sentences as similar with predictions greater than 4, and it appears this is another case of label error.

5.2 Sentiment Analysis

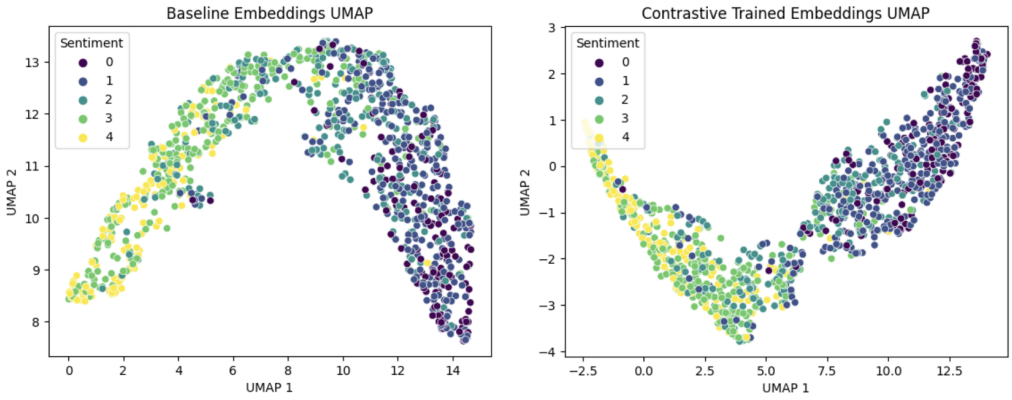

Figure 3: UMAP projections of sentence embeddings from baseline and contrastive models for the dev SST dataset.

Although the performance of baseline and contrastive pretrained models were comparable on the SST and PARA tasks, we visualized the SST sentence embeddings to investigate if we could observe meaningful changes. For both the finetune baseline and supervised contrastive models, we generated sentence embeddings for all samples in the SST dev dataset. We then performed dimensionality reduction with UMAP and plotted the projections, coloring datapoints by sentiment label. Results are shown in Figure 3.

ID	Sentence	FT Pred	Sup Pred	Label
1	Rarely has leukemia looked so shimmering and benign	4	4	1
2	It has all the excitement of eating oatmeal	2	4	1
3	It's everything you don't go to the movies for	3	4	0

Table 3: Sample of incorrect sentiment predictions from the SST dev dataset.

Although we did not observe a clear distinction between the UMAP projections of baseline and contrastive models, there is a clear separation between projections of sentences with negative and positive sentiment. For both models, there are several sentences with a very negative sentiment label and a strong positive sentiment projection neighborhood. We analyzed a sample of these sentences, shown in Table 3. For sentence 1, the models predict positive sentiment, yet the label is negative. However, "benign" indicates that the sentiment is indeed positive, so in this case, the label is incorrect. In both 2 and 3, the models predict positive sentiment but the labels indicate negative sentiment. In these cases, the models failed to identify sarcasm and therefore failed to pick up on the negative sentiment.

6 Conclusion

In conclusion, we have extended minBERT to develop a multitask model for sentiment, paraphrase, and similarity prediction. We evaluated model baselines using pretrained and finetuned model weights with a multitask round-robin training strategy. We then applied the SimCSE framework for contrastive pretraining, using both supervised and unsupervised objectives. We observed that contrastive pretraining improved performance over baseline methods, with the most significant improvement in the semantic textual similarity task. We then analyzed our baseline and contrastive model output by performing an error analysis on the similarity and sentiment tasks and observed that pretraining appears to improve sentence embeddings in some instances. However, we also observed that model performance is ultimately limited due to noisy labels and failures to identify subtle concepts such as sarcasm.

For future work, we propose extending the scope of our contrastive pretraining to include larger datasets. The unsupervised contrastive objective does not require labeled training data and can be easily applied to large volumes of unlabeled text. We also propose varying the amount of dropout to evaluate how greater degrees of data augmentation change sentence embeddings in the unsupervised objective. Finally, we propose replicating some of the experiments in the SimCSE paper to quantify the alignment and uniformity of our sentence embeddings before and after pretraining.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805 [cs].

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. SimCSE: Simple Contrastive Learning of Sentence Embeddings. ArXiv:2104.08821 [cs].

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. ArXiv:1412.6980 [cs].

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. ArXiv:1301.3781 [cs].

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. ArXiv:1706.03762 [cs].