BERT's Multitask Learning Adventures

Stanford CS224N Default Project

Jonathan Larkin
Stanford University
jonathan.r.larkin@gmail.com

Yipeng Liu
Stanford University
yipengl@stanford.edu

Abstract

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that has shown impressive results in various NLP tasks such as sentiment analysis, question answering, and text classification [1]. In this project, we explore the effectiveness of *multitask learning* with BERT, where the model is trained on multiple NLP tasks simultaneously. We explore how a single BERT model can learn sentiment classification (SST), paraphrase classification (PARA), and text similarity (STS) detection. We employed data augmentation, contrasting pretraining, "gradient surgery," data resampling, and a model architecture with a single BERT backbone and a shared layer for PARA and STS. Our experiments show that a single BERT model can effectively learn multiple tasks.

1 Key Information to include

- External collaborators (if you have any): None
- External mentor (if you have any): None
- Sharing project: N/A

2 Introduction

It is shown that BERT (Bidirectional Encoder Representations from Transformers) [1] can achieve impressive performance when fine-tuned on various downstream tasks. Intuitively, during the fine-tuning process the BERT model will learn important features from the training data that are helpful to the specific task.

In this project, we explored a more interesting question: How effective are the learned embeddings? Will a single BERT be able to produce universal contextual embeddings of sentences that can be used in multiple downstream tasks? This is an important question to ask, since a positive answer would imply that BERT is powerful enough to learn multiple possibly unrelated features at the same time. Having such a general-purpose model can greatly help with time and space efficiency in NLP tasks. In particular, we investigated how the same embeddings can be used in the three downstream tasks sentiment classification (SST), paraphrase classification (PARA), and text similarity detection (STS). While PARA and STS are similar in nature, they do not share much in common with SST. Therefore, being able to perform well simultaneously on the three tasks means that BERT indeed has the capability to learn universal contextual embedddings.

3 Related Work

In the original BERT paper [1] in 2018, Devlin et al. fine-tuned BERT independently on all the GLUE [2] tasks and achieved an averge GLUE score of **80.5** on the official GLUE leaderboard [1] on the BERT$_{LARGE}$ model. Then in 2019, Liu et al. [3] proposed the MT-DNN model which is a

[1] https://gluebenchmark.com/leaderboard

multitask BERT-based model that pushes the GLUE benchmark to **82.7**. This improvement implies that training BERT on multiple tasks simultaneously can actually help the model understand natural languages better in general and helps with the performance on the individual tasks. This has further inspired more works on optimizing multitask training, including T5 [4], SMART [5], and the current state-of-the-art [2] XY-LENT [6] with a score of **91.3**.

Some of the important optimizations that we have implemented in our model are inspired by previous works, including Gradient Surgery [7] which deals with conflicting gradients in different tasks, unsupervised SimCSE [8] which improves the embeddings through additional pretraining, and the "IMDB 50k" dataset [9] which provides additional training data for the SST task.

4 Approach

As required, we first implemented the BERT architecture and AdamW optimizer. The architecture and the optimization algorithm are described in detail in the project handout, so we will focus on the multitask classifier in this section. In the following subsections, we coded everything by ourselves unless otherwise specified.

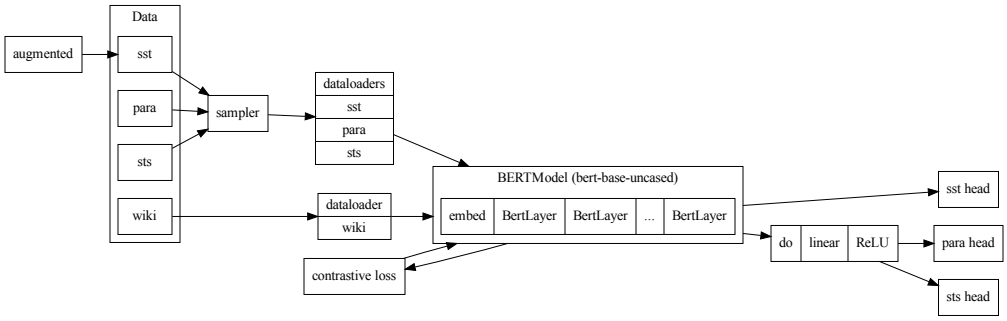


Figure 1: Multitask Model Architecture

4.1 Baseline Model

Our baseline model is simply the BERT model followed by three custom heads, one for each task (sentiment classification, SST, paraphrase classification, PARA, and similarity regression, STS). The architecture of each head is a dropout layer followed by a linear layer. In all cases, we used the pooled representation of the input, which is the final hidden state for the `[CLS]` token after a forward pass through the BERT backbone, as the contextual embedding for the sentence. For PARA and STS, we pass each input separately through the BERT backbone and then concatenate the embeddings for the two sentences to get an embedding of size $2 \times$ `hidden_size`. This concatenated vector is then passed through the dropout and linear linear. Finally during training, we train the three heads sequentially in each epoch and sum up the losses.

4.2 Concatenation of Sentences

First, we noticed that while SST and PARA achieve reasonable performance in the baseline model (compared to other groups on the “dev leaderboard"), STS performed poorly. We were able to make improvements by referring to Figure 2 in the appendix from the original BERT paper [1]. Instead of feeding each sentence into BERT and concatenating their `CLS` embeddings, we now do only one forward pass by combining the two sentences into one. Specifically, we will strip out the [CLS] token of the second sentence and add a [SEP] token to get a single input like this:

[2] As of March 15, 2023.

`[CLS] id1_1 id1_2...[PAD]...[PAD] [SEP] id2_1 id2_2...[SEP] [PAD] [PAD]...`
We believe that this modification will help with the performance because instead of having two independent contextual embeddings from BERT, we are now treating the two sentences as a whole and forcing BERT to learn one embedding that captures the relationship between the two sentences, which will be helpful in the downstream STS head.

In addition, we realized that the same reasoning could also be applied to PARA. In fact, given the similar nature of STS and PARA, it is reasonable to let them have the same head architecture so that they will be able to help with each other's learning.

4.3 Multitask Training with Data Sampling

As described in the Data section below, the three datasets, SST, PARA, and STS are all of different sizes. This presents a challenge for model fitting since, for a single batch size, the number of batches differs per dataset. In a *preprocessing* step, we took a *data resampling* approach to create training datasets of equal sizes. For computational reasons we limited the size of each dataset to 20,000 samples. For datasets with more than 20,000 samples, we randomly selected (without replacement) 20,000 samples; for datasets with less than 20,000 samples, we repeatedly sampled the dataset (with replacement) until we built a dataset of 20,000 samples. The result of this preprocessing step was that each dataset now contained the same number of batches. Following Bi et al. [10], for multitask training, we took a batch from each dataset for a single iteration, calculated a loss per dataset, added the losses, and then backpropagated.

4.4 Shared Layer

We observed that the PARA and STS tasks were similar: in both cases we want the model to learn how similar two segments of text are. To aid the model to capture this fundamental similarity, we added a shared expressive layer after the BERT backbone pooling output and before the PARA and STS heads. As shown in Figure 1, this layer consists of a dropout layer, a linear layer, and a ReLU activation.

4.5 Gradient Surgery for Multitask Fine-Tuning

When training under the finetune option, the parameters in the BERT model is unfrozen and will be updated for every batch. As described in the multitask training subsection, we take a batch from all three datasets and sum up the training loss. We then use gradient descent with this total loss. However, it is possible that the directions of the three individual gradients are in conflicting directions. As a result, learning them all at once could lead to worse overall performance and hurt data efficiency. Yu et al. [7] proposed a general approach to avoid such interference between multitask gradients called Gradient Surgery. It projects the gradient of a task onto the normal plane of the gradient of another task that has a conflicting gradient. Specifically, it computes the gradient magnitude similarity of two gradients

$$\Phi(\mathbf{g}_i, \mathbf{g}_j) = \frac{2\|\mathbf{g}_i\|_2\|\mathbf{g}_j\|_2}{\|\mathbf{g}_i\|_2^2 + \|\mathbf{g}_j\|_2^2}$$

and uses it to decide whether they are conflicting. If the similarity if negative, which means they two gradients are conflicting, then we will replace $\mathbf{g}_i$ to be

$$\mathbf{g}_i = \mathbf{g}_i - \frac{\mathbf{g}_i \cdot \mathbf{g}_j}{\|\mathbf{g}_j\|^2}\mathbf{g}_j.$$

Otherwise $\mathbf{g}_i$ is not changed. We used the implementation of Gradient Surgery by Tseng [11] in our code.

4.6 Contrastive Pretraining

We seek to improve the quality of the contextual embedding produced by BERT through additional pretraining under a contrastive learning framework. Specifically, we implemented the unsupervised SimCSE proposed by Gao et al [8]. Figure 3 in the appendix gives a visualization of how it works. Given a batch in the wiki dataset, we feed every sentence in it into the BERT model twice. The

dropout layers in BERT will act as noise and produce two different but hopefully similar contextual embeddings of the same sentence by taking the encodings of the `[CLS]` tokens. Then, for every sentence we ask the model to predict only itself. To do so, we let the other embedding of the same sentence be the only positive while all other embeddings be in-batch negatives. Then we apply the cross entropy loss function on the cosine similarities between the sentence and all other sentences in the batch. In other words, with a set of paired embeddings $\{(x_i, x_i^+)\}_{i=1}^n$ where n is the batch size, the training objective for (x_i, x_i^+) is

$$J = \texttt{cross\_entropy}((sim(x_i, x_1^+), sim(x_i, x_2^+), \cdots, sim(x_i, x_n^+)), \mathbf{v}_i)$$

where sim is the cosine similarity and $\mathbf{v}_i$ is the one-hot vector that represents the index i.

4.7 Data Augmentation

As noted in the Data section on SST below, the SST dataset contains only 8,544 samples. It is well known that (a) deep learning models, *ceteris paribus*, show improved performance with additional data [cite?], and (b) *data augmentation* can improve performance of deep learning models when data is small and help reduce overfitting [cite?]. We utilized two data augmentation techniques to more than double the size of dataset. The first technique is *generative paraphrasing*: we used the **OpenAI GPT-3 API** and called the `text-davinci-003` engine with the following prompt for each sentence in SST: "Write a paraphases of "{sentence}" ". We took each output sample, labeled with the sentiment from the input sentence, as a new sample. This doubled the size of the SST dataset. Three examples of generative paraphrasing augmentation are in Table 1.

sentence	augment	sentiment
The Rock is destined to be the 21st Century 's new " Conan " and that he 's going to make a splash even greater than Arnold Schwarzenegger , Jean-Claud Van Damme or Steven Segal .	Dwayne Johnson is sure to be remembered as one of the greatest action stars of the 21st Century, much like Arnold Schwarzenegger, Jean-Claud Van Damme, and Steven Segal were in the previous century. He is predicted to make an even bigger impact than them.	3
The gorgeously elaborate continuation of " The Lord of the Rings " trilogy is so huge that a column of words can not adequately describe co-writer\/director Peter Jackson 's expanded vision of J.R.R. Tolkien 's Middle-earth .	Peter Jackson's magnificently detailed version of the "The Lord of the Rings" trilogy is so broad that it can't be explained in a single sentence. His interpretation of J.R.R. Tolkien's Middle-earth is so expansive that it must be experienced to be truly appreciated.	4
Singer\/composer Bryan Adams contributes a slew of songs – a few potential hits , a few more simply intrusive to the story – but the whole package certainly captures the intended , er , spirit of the piece .	Bryan Adams brings a wide variety of songs to the project - some of which are sure to be hits, while others are more subtle - but all of them come together to give the work the feeling it was meant to have.	3

Table 1: SST Generative Paraphrasing Augmentation Examples.

The second technique we used for augmentation is *back translation*. For this technique, we took each sentence in the SST dataset and passed it into the **Google Translate API** calling for translation from English to Spanish. We then passed the Spanish translation back into the Google Translate API calling for translation from Spanish to English. These two passes generate sentences with slight variation to the original SST sentences. In many cases, the back translation comes out with an *exact* or very close match. In order to screen out matches that were exact duplicates or "too close", we filtered the output. To filter the output we created a single-word n-gram count vector (with a single global vocabulary) for each source sentence and each back-translated sentence. For each pair of sentences, we calculated the cosine similarity and discarded back-translated sentences where it's cosine similarity was greater than 0.80. After this filter we were left with 3,132 new sentences. Three examples are in Table 2.

After adding in the augmentations, the number of samples in our SST dataset grew from 8,544 to 20,220. In a simple ablation study, we trained a BERT model with a single sentiment head on the

original data, and then on the fully augmented data. Training on the augmented data leads to an improvement in accuracy on the dev from 0.521 to 0.526.

sentence	augment	sentiment
The Sundance Film Festival has become so buzz-obsessed that fans and producers descend upon Utah each January to ferret out The Next Great Thing .	The Sundance Film Festival has become so obsessed with rumors that fans and producers alike flock to Utah every January to find out about The Next Great Thing.	2
They are what makes it worth the trip to the theatre .	They are what make the trip to the theater worthwhile.	3
Light , silly , photographed with colour and depth , and rather a good time .	Light, silly, photographed with color and depth, and quite funny.	4

Table 2: SST Back-Translate Augmentation Examples.

5 Experiments

5.1 Data

SST. We used the Stanford Sentiment Treebank (SST) [12] provided in the starter files.

PARA. We used the SemEval STS Benchmark Dataset [13] provided in the starter files.

STS. We used a sub-sample from the Quora Dataset since the original dataset is large (141K examples) and makes training slow. In addition, we noticed that the Quora Dataset is imbalanced in terms of the number of the two labels and we also fixed this during sampling. For the baseline training, we are using a dataset with 10,000 "0" labels and 10,000 "1" labels.

Wiki Sentences. This dataset contains 1 million sentences randomly sampled from English Wikipedia, and is used to perform additional contrastive pretraining on the BERT model. We used the same dataset as in the SimCSE paper [8].

IMDB 50k. For some experiments, we utilized the “IMDB 50k" dataset [9]. The fine-grained SST training dataset as given contains only 8,544 samples. There are many other “sentiment" datasets in the wild, such as IMDB 50k, however they are binary: the labels are “positive" or “negative". We theorized that adding an *additional* task of predicting binary sentiment on the IMDB 50k dataset could improve the multitask model’s ability to predict the SST fine-grained sentiment. We added an IMDB data loader and ran experiments including (a) adding an IMDB binary head and adding that task into the multitask training, (b) adding a “dropout/linear/ReLU" layer shared between the IMDB binary head and the SST fine-grained head, (c) adding a full new BERT Layer shared between the IMDB and SST heads, and (d) pretraining the BERT Layer and IMDB head on the IMDB 50k dataset. We had expected that using the IMDB dataset would dramatically increase the SST performance because (a) the IMDB dataset is large and (b) learning binary sentiment classification should have incremental benefit to fine-grained sentiment prediction. None of these experiments, however, showed increased performance on the dev sets. We did not use the IMDB dataset in the final submission.

5.2 Evaluation method

For both sentiment analysis and paraphrase detection, we used accuracy as the evaluation metric with the predictions and true labels being integers (0-4 for sentiment analysis and 0-1 for paraphrase detection). For semantic textual similarity, we evaluated the results using the correlation between the predictions (real numbers from 0 to 1) and the true labels (real numbers from 0 to 5). It is appropriate since correlation is invariant under constant scalar.

5.3 Experimental details

The model architecture for our submitted results (the “multitask-final" run) is shown in Figure 1. We first ran single-task contrastive finetuning on 100,000 Wiki sentences for 10 epochs with a learning

rate of 1e-5. The model file for that run was then loaded as a checkpoint and we ran the multitask training, employing “gradient surgery" and sampling the SST, PARA, and STS datasets to 20,000 samples at a learning rate of 1e-5 for 10 epochs. We employed early-stopping on the post epoch dev set performance. In total we ran over 100 experiments, tracked with Weights Biases. We tried contrastive pre-training on one million wiki sentences, different architectures (as described in the section 5.1 “IMDB 50k" above), a smaller learning rate, and smaller and larger dropout rates. Those experiments did not show improvement over the “multitask-final" run.

5.4 Results

In table 3 we show the progression in prediction scores from the baselines and milestone report. We were pleased to see that our model made substantial improvements as we progressed with our experiments. We note that the scores on “dev" set, not shown in the table, for “multitask-final" were 0.51, 0.841, 0.845 for SST, PARA, and STS respectively.

dev acc/corr	SST	Paraphrase	STS
first-baseline-pretrain	0.351	0.626	0.241
first-baseline-finetune	0.493	0.729	0.291
milestone-baseline-pretrain	0.310	0.662	0.417
milestone-baseline-finetune	0.499	0.725	0.555
test acc/corr	**SST**	**Paraphrase**	**STS**
multitask-final	**0.521**	**0.832**	**0.801**

Table 3: Baseline, Milestone, and Final Results.

6 Analysis

We performed “error analysis" for the final multitask model by looking at the predictions on the dev set versus the ground truth labels. The confusion matrices for the STS and PARA tasks, and a scatter plot for the STS task are shown in Figure 2.

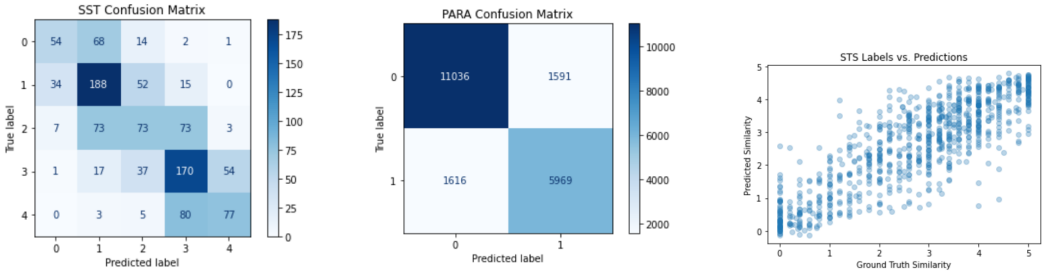


Figure 2: Multitask Model Error Analysis

For SST, we note that the actual sentiment differences between the human labeled ground truth between close labels (e.g., 0 and 1) are fairly subjective. As such, we would not expect a model to perform extremely well in prediction. We investigated the cases of *extreme* errors: i.e., predicting 4 versus a ground truth of 0, or vice versa. The sentence where the model predicts 4 and the label is actually 0 is “It 's everything you do n't go to the movies for ". Here we see that the single token “n't" flips the sentiment of the sentence. It is disappointing that the model does not correctly predict this sample as this failure mode is very common in simpler n-gram-style bag of words models. The value proposition, in part, of transformer-based models is to capture the full context and not make these errors. There are no samples where the model predicts 0 versus a ground truth of 4. The “worst" prediction in this direction is the sentence where the model predicts 0 but the ground truth is 3. This sentence is “It seems like I have been waiting my whole life for this movie and now I ca n't wait for the sequel". It is not obvious why the model fails here.

To investigate the performance of PARA, we sample five false positives and five false negatives. Our review of these “errors" (tables 4 and 5) indicates that they may in part be errors of ground truth

labels. For example, the sentences “Will the Kashmir issue ever end?" and “Will there ever be an end to the Kashmir issue?" are labelled as “not duplicate".

sentence1	sentence2	is_duplicate	predicted
Will the Kashmir issue ever end?	Will there ever be an end to the Kashmir issue?	0	1
How do I hide my interests in Tinder?	How do I delete my interests on tinder?	0	1
Can hummingbirds fly backwards?	How do hummingbirds fly backwards?	0	1
How should one get rid of chronic dandruff?	How do you get rid of severe dandruff?	0	1
Which is the one movie scene you watch again and again?	What is the best movie that you have no desire to watch again?	0	1

Table 4: PARA Sample of False Positives

sentence1	sentence2	is_duplicate	predicted
Am I forced to say the pledge of allegiance at school when I have a different religion?	Can my school legally force me to say the pledge of allegiance, and punish me if I refuse?	1	0
How can you describe the eukaryotic cell cycle?	What is the eukaryotic cell cycle?	1	0
What is a double taxation agreement?	What does double taxation mean?	1	0
How many views and answers are required to become Top Writer in Quora?	How do you become a Top Writer on Quora for 2014?	1	0
Are there any hacks for Shadow Fight 2?	How do I hack Shadow Fight 2?	1	0

Table 5: PARA Sample of False Negatives

Lastly, for STS, since this is a regression task, we look at the plot of the predicted values versus the ground truth. We query to show the two samples where the ground truth similarity is greater than four, but the predicted similarity is less than one. These two samples are in Table 6 and they seem like genuine prediction errors. In the first case it seems the model is confused by the different methods of negation and the semantic similarity of “general answer" and "single definition." In the second case, the model is confused about the domain-specific financial language.

sentence1	sentence2	similarity	predicted
I think there isn’t a general answer.	I don’t think there is a single definition.	4.0	0.759903
Stock index futures point to lower start	Stock index futures signal early losses	4.4	0.956338

Table 6: PARA Sample of False Negatives

7 Conclusion

This project presented many interesting challenges, both conceptually and in implementation/engineering. We delved into the inner workings of the famous BERT model, built pipelines for pre-training and fine-tuning, and adapted the model for three tasks. While some ideas we implemented, such as gradient surgery, contrastive pretraining, and data augmentation indeed helped the

model, we were disappointed that other ideas, such as using the IMDB50k dataset to aide in SST prediction, did not help. There are many other experiments we would have liked to try, including "Multiple Negative Ranking Loss" and more exhaustive hyperparameter tuning. Ultimately we were limited by the timeline of the project. The error analysis highlights a common problem in assessing NLP models on human labeled data: the labels may be ambiguous or "wrong" in some cases. This projected highlighted the complexity of utilizing a single model for multiple tasks.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[2] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461, 2018.

[3] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding, 2019.

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[5] Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online, July 2020. Association for Computational Linguistics.

[6] Barun Patra, Saksham Singhal, Shaohan Huang, Zewen Chi, Li Dong, Furu Wei, Vishrav Chaudhary, and Xia Song. Beyond english-centric bitexts for better multilingual language representation learning, 2022.

[7] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.

[8] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings, 2021.

[9] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

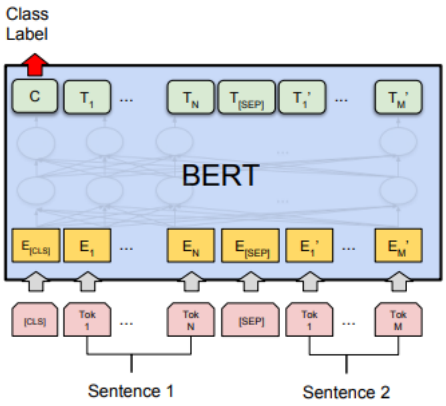
[10] Qiwei Bi, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Hanfang Yang. MTRec: Multi-task learning over BERT for news recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2663–2669, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[11] Wei-Cheng Tseng. Weichengtseng/pytorch-pcgrad, 2020.

[12] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[13] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. *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, June 2013. Association for Computational Linguistics.

A Appendix



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

Figure 3: Fine-tuning sentence pair tasks in BERT

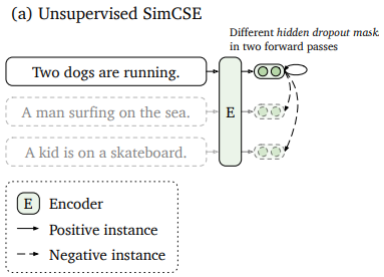


Figure 4: Unsupervised SimCSE